Xtext

introductory workshop

Holger Schill, Stefan Oehme, Moritz Eysholdt
itemis AG

Installation

- Grab a USB Key
- Install Eclipse
- Save the Zip Files to Your Disk
  - host_workspace.zip
  - runtime_workspace.zip

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

9:30 - 10:45	11:00 - 12:00	Lunch	13:00 - 14:30	Coffee	15:00 - 16:30

Outline

- **Example Application**
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

Eclipse Community Survey 2013

Eclipse Community Survey 2013

Eclipse Community Survey 2013 Evaluate

What is the *primary* computer language you typically use to develop software?

- C/C++
- C
- Lua
- Groovy
- Java
- Java Script

Demo

- Ruby
- Scala
- Visual Basic/Visual Basic .Net
- None - I don't use a computer language for software development
- Don't know
- Other (please specify) Xtend

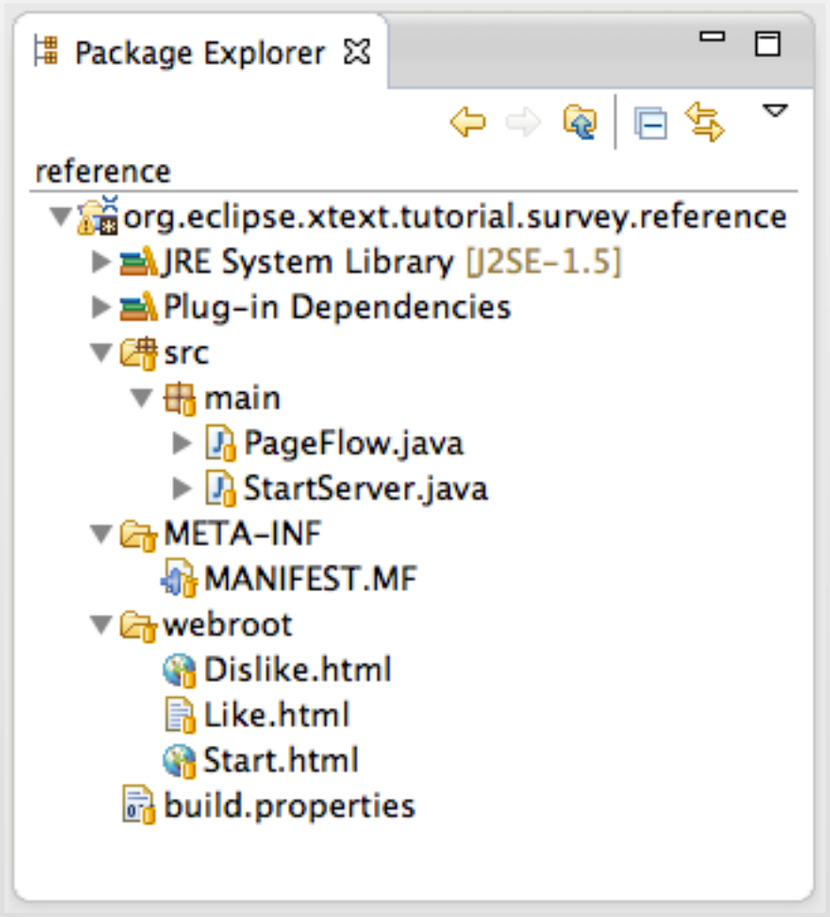
Example: Surveys

- A Web-Based App for Surveys
  - Online Answering
  - Multiple Pages
  - Different Types of Questions
  - Online Evaluation

Architecture

- HTML Forms
- Twitter Bootstrap CSS / JavaScript
- Jetty Server
- Simple Page-Flow Engine
- In-Memory Persistence

API View


Package Explorer
reference
org.eclipse.xtext.tutorial.survey.reference
JRE System Library [J2SE-1.5]
Plug-in Dependencies
src
main
PageFlow.java
StartServer.java
META-INF
MANIFEST.MF
webroot
Dislike.html
Like.html
Start.html
build.properties

Challenges

- A Heterogeneous Platform
  - Java, HTML, Database ...
- Difficult to Extend
  - More Questions, Other Surveys
  - Other Front-Ends (iOS, Android, PDF, ...)
- Hard to Maintain

The Domain

- The application is about **Surveys**.
- A **Survey** consists of **Pages**.
- A **Page** holds **Questions**.
- **Questions** are answered with **FreeText** or predefined **Choices**.
  - Some **Choices** are **exclusive**.
- A **Page** defines its **FollowUp** pages
  - **FollowUps** may depend on given answers.

DSL Approach

- Create a Domain-Specific Language
- Describe the Data Formally
- Generate Code from DSL Files

Survey

Page
- **TextQuestion**
- **ChoiceQuestion (single)**
  - **Choice**
  - **Choice**
- **FollowUp**
- **FollowUp**

```
survey tutorial "EclipseCon 2013 Tutorial Survey"

page Start (
    text name 'Your name'
    single choice like "Do you like the tutorial?" (
        yes "Yes"
        no "No"
        wat "Which tutorial? I'm waiting for the bus!"
    )
  if like=yes -> Like
  if like=no -> Dislike
)
```

Page
- **ChoiceQuestion**
  - **Choice**
  - **Choice**
  - **Choice**

```
page Like (
    choice particular "What do you like in particular?" (
        xtext 'Xtext is awesome'
        excercises 'The funny exercises'
        tutors 'The handsome tutors'
    )
)
```

Page
- **ChoiceQuestion**
  - **Choice**
  - **Choice**
  - **Choice**

```
page Dislike (
    choice particular "What do you hate in particular?" (
        xtext 'Xtext sucks'
        excercises 'The boring exercises'
        tutors 'The tutors stink'
    )
)
```

Advantages

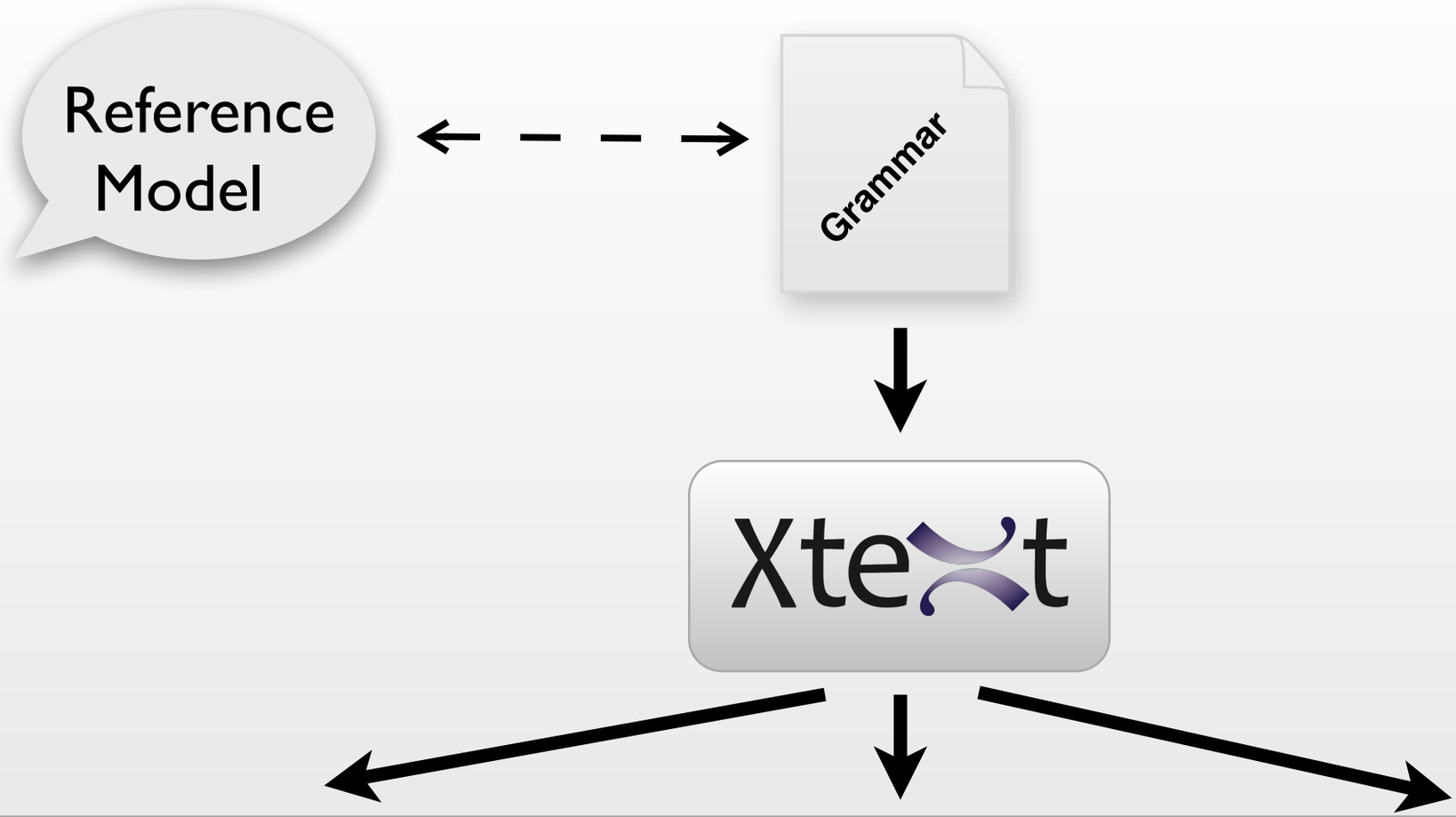
- Addresses Heterogeneity
- Easy Design of Surveys
- Easy to Add New Front-Ends
- Separation of Roles During Development
- Speed-up for Development
- Improved Maintainability

Outline

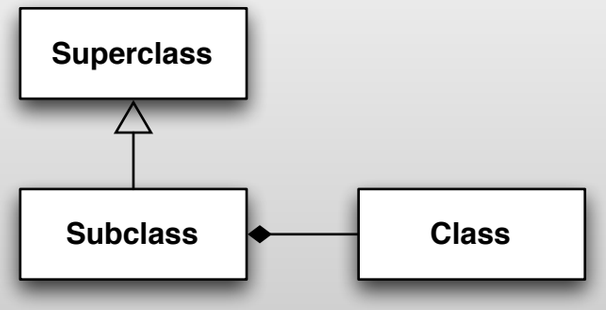
- Example Application
- **Build your DSL with Xtext**
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

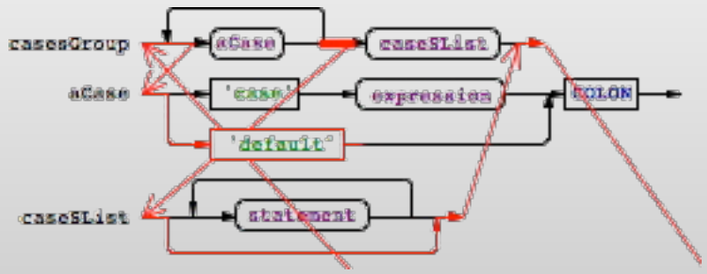
9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30



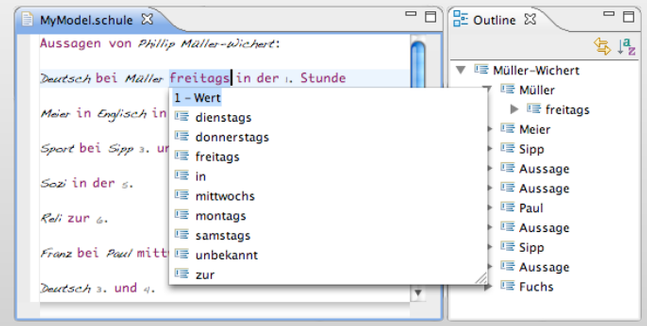
Xtext Runtime



Ecore Package

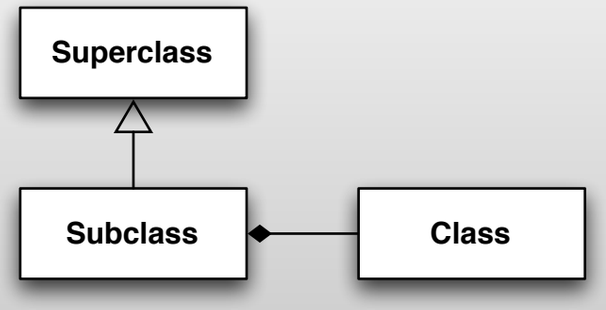


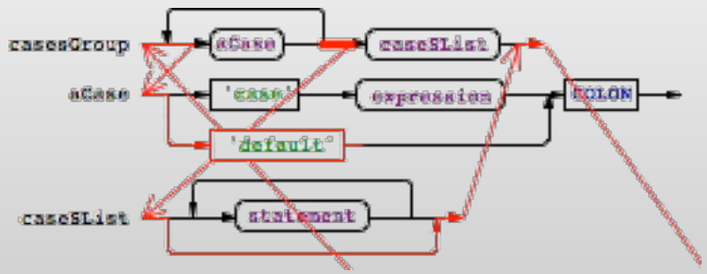
Parser

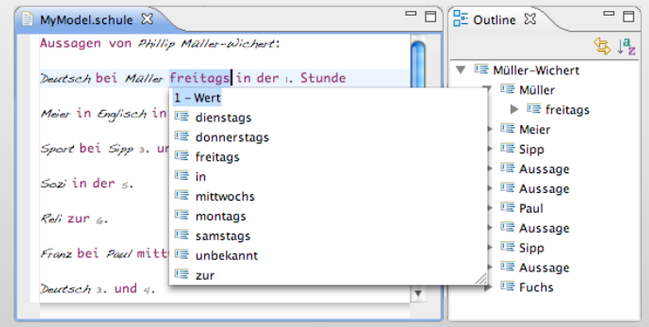


Editor

Xtext Runtime







Ecore model

LL(*) Parser

Editor

<uses types>

<instanceof>

<instantiates>

<parsed by>

<creates>

Interpreter

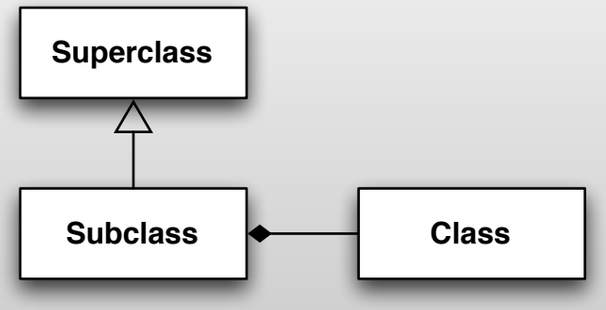
Code Generator

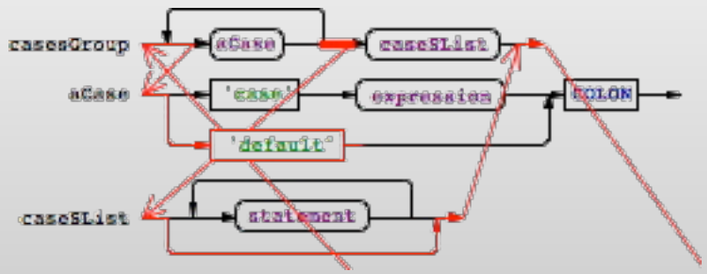
<processed by>

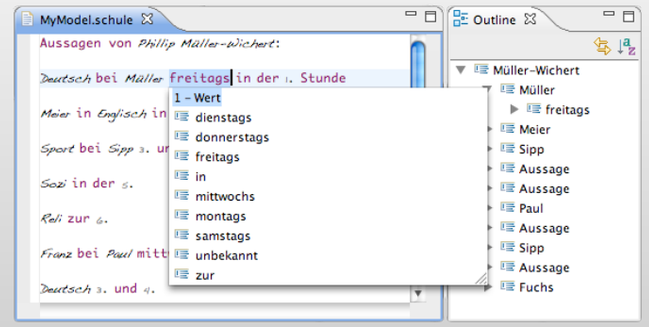
AST

Textual Model

Xtext Runtime







Ecore model

Serializer

Editor

<uses types>

<instanceof>

<input for>

<creates/updates>

<populates>

Quickfix

Refactoring

Graphical Editor

<processes>

AST

Textual Model

EXERCISE:
NEW PROJECT

Grammar Definition

Grammar Reuse

Derived Metamodel

Multivalue Assignment

Implicit return type *Model*

Cardinality (one or more)

Parser Rules

Alternative

Implicit return type *Type*

Simple Assignment

Attribute

Boolean Assignment

Optional (zero or one)

Cardinality (zero or more)

Containment Reference

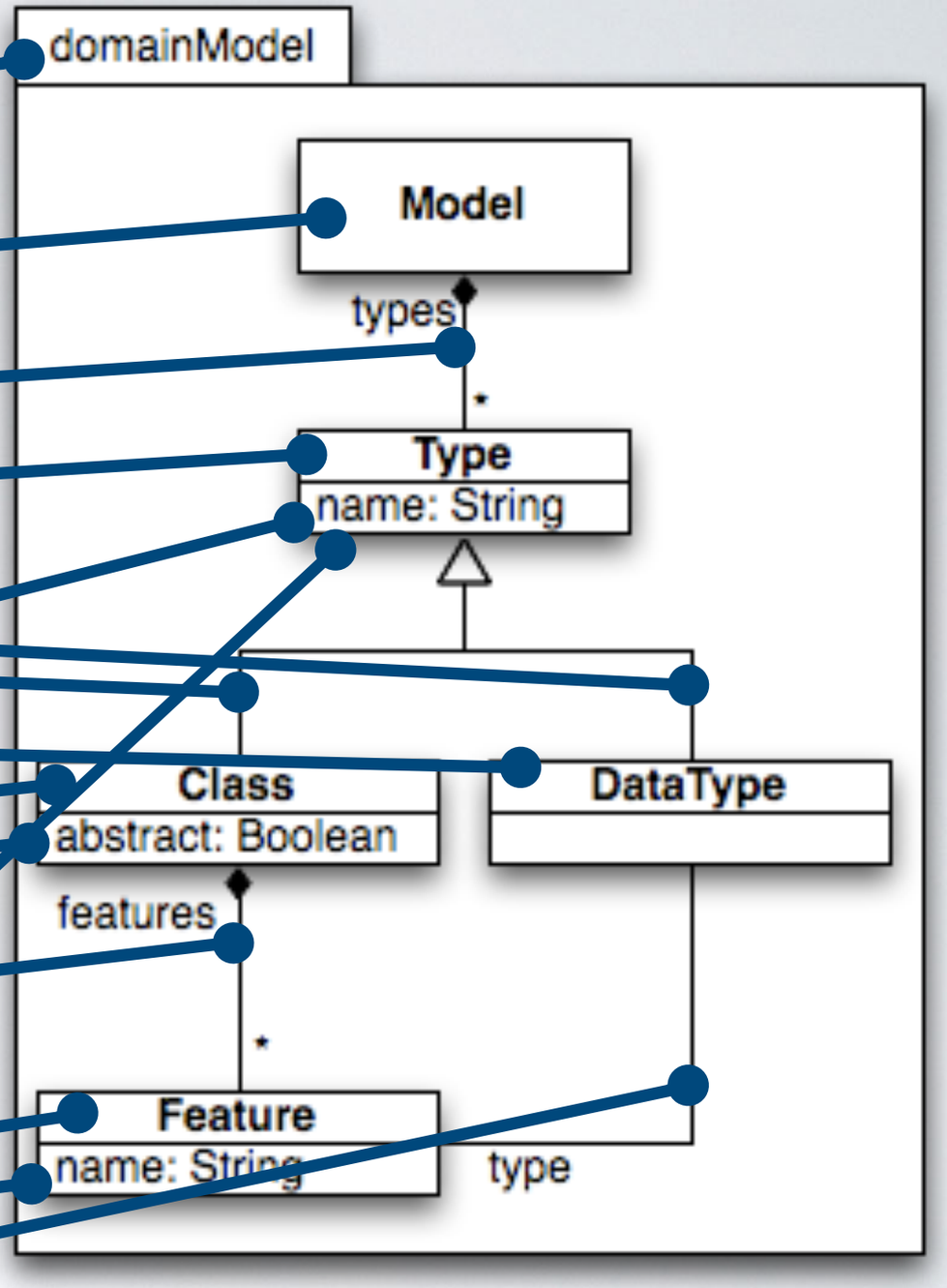
Keywords

Cross Reference

```
grammar org.xtext.workshop.DomainModel
    with org.eclipse.xtext.common.Terminals

generate domainModel "http://www.xtext.org/workshop/DomainModel"

Model:
    (types+=Type)+;

Type:
  Class | DataType;

DataType:
  'datatype' name=ID;

Class:
    (abstract?='abstract')? 'class' name=ID '{'
    (features+=Attribute)*
  '}';

Attribute returns Feature:
  'attr' name=ID ':' type=[DataType];
```

```
dataType String
abstract class Person {
    attr name : String
}
```

```
grammar org.xtext.workshop.DomainModel
    with org.eclipse.xtext.common.Terminals

generate domainModel
  "http://www.xtext.org/workshop/DomainModel"

Model:
    (types+=Type)+;

Type:
  Class | DataType;

DataType:
  'datatype' name=ID;

Class:
    (abstract?='abstract')? 'class' name=ID '{'
    (features+=Attribute)*
  '}';

Attribute returns Feature:
  'attr' name=ID ':' type=[DataType];
```



EXERCISE:
GRAMMAR

Outline

- Example Application
- Build your DSL with Xtext
- **Generate Code with Xtend**
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30


Xtend

Hello Xtend

```
class Greeter {
  def static void main(String... args) {
    println('Hello EclipseCon')
  }
}
```

Templates

```
def example(List<String> elements) '''
   Usually a template consists mainly of text spanning
    multiple lines.
  If you want to evaluate an expression you have to write it
  in french quotes «7*3*2». Code assist inserts a pair of
  these.
  You can also iterate a collection with FOR
    «FOR element: elements»
    Found another element: «element»
    «ENDFOR»
  For decisions there is the IF statement
  «IF elements.empty»
        no elements.
    «ENDIF»
'''
```

Dispatch Methods

```
// Xtend code
def dispatch area(Circle c) {
   c.radius * c.radius * Math.PI
}
def dispatch area(Rectangle r) {
   r.width * r.height
}


def someCalculation(Object o) {
   area(o) // polymorphic call
}
```

```
// generated Java code (dispatcher)
public double area(final Object c) {
  if (c instanceof Circle) {
    return _area((Circle)c);
  } else if (c instanceof Rectangle) {
    return _area((Rectangle)c);
  } else {
    throw new IllegalArgumentException();
  }
}
```

EXERCISE:
CODE GENERATOR

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- **Validate Models**
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

Validator

```
class SurveyValidator extends AbstractSurveyValidator {
    @Check
    def textMustNotBeEmpty(Question question) {
        if(question.getText().isEmpty()) {
            error("Empty question is illegal",
              question,
              SurveyPackage.Literals.QUESTION__TEXT)
        }
    }
}
```

EXERCISE:
VALIDATOR

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- **Cross-References and Scoping**
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

Cross References

```
page Start (
    single choice like "Do you like the tutorial?" (
        yes "Yes"
        no "No"
    )
    if like=yes -> Like
)

page Like (
    choice particular "What do you like in particular?" (
        xtext 'Xtext is awesome'
        excercises 'The funny exercises'
        tutors 'The handsome tutors'
    )
)
```

Grammar

```
Page:
   'page' name=ID '('
   // questions
      '->' next=[Page|ID]
')';
```

WHAT'S A SCOPE?

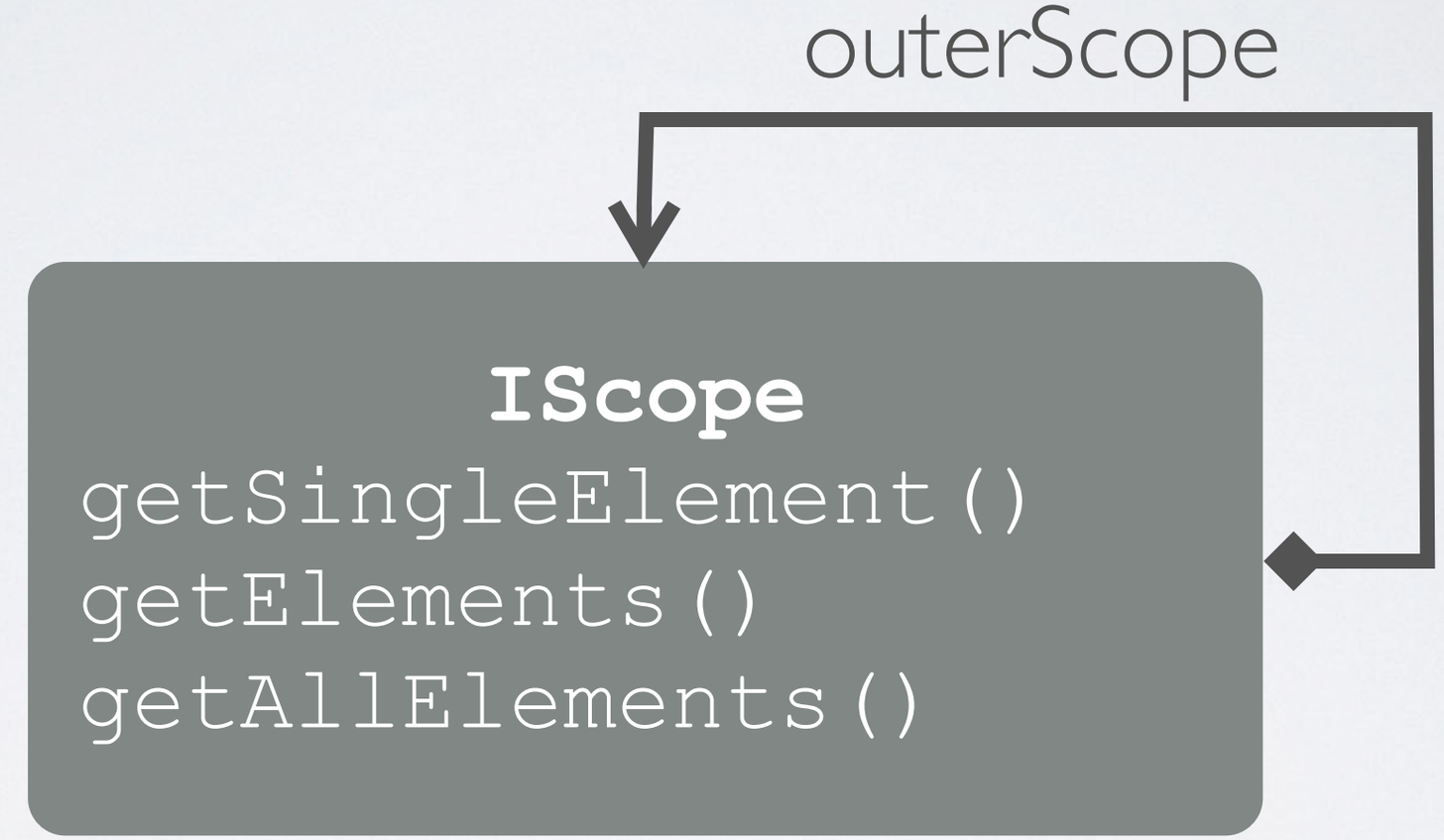
Describes the visible Elements

Depends on the context

Mapping: Name to EObject

Chain of Responsibility

WHAT'S A SCOPE?



CROSS REFERENCES

```
...
<children xsi:type="application:PerspectiveStack"
          xmi:id="_NGiG4DGjEd-zge_czUaGVQ"
          id="_NGiG4DGjEd-zge_czUaGVQ"
          selectedElement="_NGjVADGjEd-zge_czUaGVQ">
   <children xsi:type="application:Perspective"
             xmi:id="_NGjVADGjEd-zge_czUaGVQ"
             id="_NGjVADGjEd-zge_czUaGVQ">
...
```

UUIDs

Names

```
PerspectiveStack :myDslPerspectiveStack
   selectedElement=myDslPerspective
...

  Perspective :myDslPerspective
...
```

QUALIFIED NAMES

```
application contacts {
    command save "Save"
}
```

EAttribute „name“
EAttribute „name“

qualifiedName(I)

Default
QualifiedNameProvider

contacts. save

OBJECTS CAN HAVE MULTIPLE DIFFERENT NAMES DEPENDING ON THE CONTEXT

SCOPES

```
application contacts {
    command save "Save"
    command exit "Exit"
    bind "CTRL+S" to save
    bind "CTRL+X" to contacts.exit
}
```

1 2 3

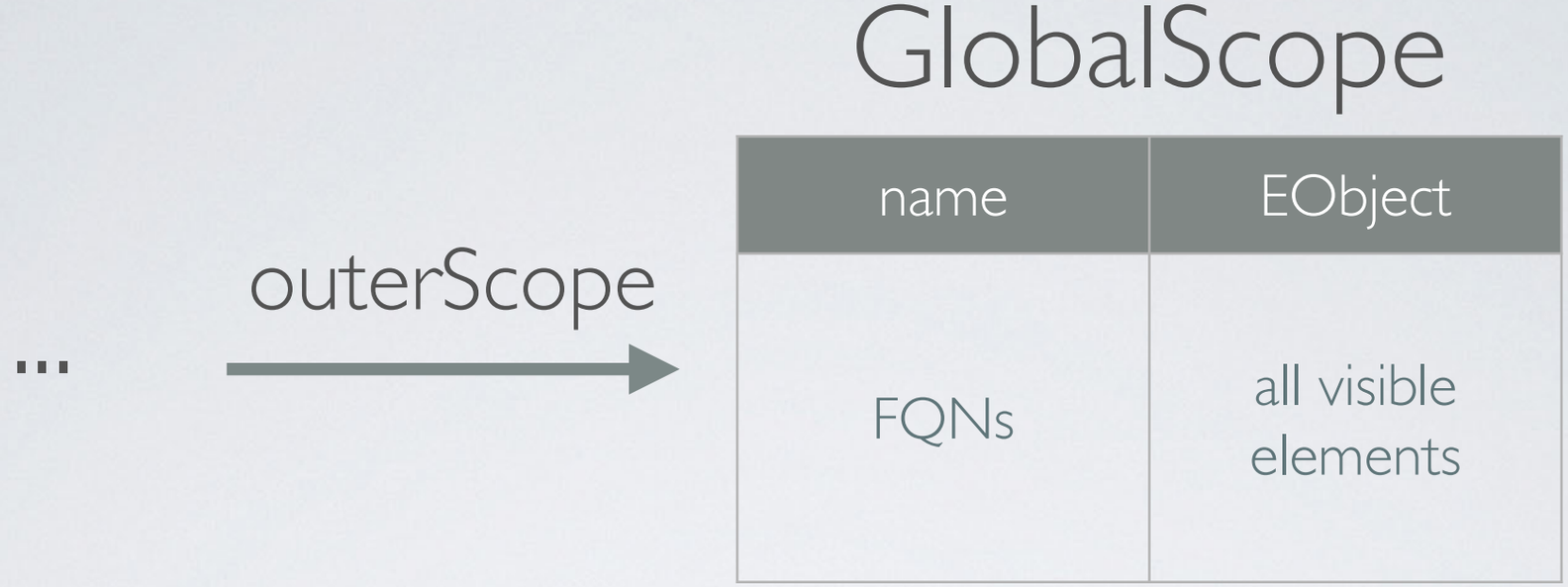
Container's Scope

name	EObject
,,contacts“	1
,,save“	2
,,exit“	3

outerScope

File Scope

name	EObject
,,contacts“	1
,,contacts.save“	2
,,contacts.exit“	3



DefaultGlobalScopeProvider

- all in workspace
- backed by an Eclipse builder

ResourceSetGlobalScopeProvider

- all in ResourceSet

Scoping

```
page Start (
    text name 'Your name'
    single choice like '...' (
        yes 'Yes'
        no 'No'
    )
    if like=yes -> Like
)

page Like (
    choice particular '...' (
        xtext '...'
        excercises '...'
        tutors '...'
    )
...
```

Questions (Default Scoping)	
Global Name	Local Name
Start.name	name
Start.like	like
Like.particular	

Choices
Customized
yes
no

EXERCISE:
SCOPING

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- **Advanced Grammar**
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

GRAMMAR LANGUAGE - ADVANCED FEATURES

```
terminal ID      : '^'?('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'_'|'0'..'9')*;
terminal INT returns ecore::EInt: ('0'..'9')+;
```

```
QualifiedName:
    ID ('.'  ID)*;

Double returns ecore::EDouble:
    INT ('.' INT)?;
```

```
enum Visibility:
    public="public" | private="private" | protected="protected";
```

GRAMMAR LANGUAGE - ADVANCED FEATURES

```
terminal ID      : '^'?('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'_'|'0'..'9')*;
terminal INT returns ecore::EInt: ('0'..'9')+;
```

```
QualifiedName:
    ID ('.' ID)*;

Double returns ecore::EDouble:
    INT ('.' INT)?;
```

```
enum Visibility:
    public | private | protected;
```

GRAMMAR LANGUAGE - ADVANCED FEATURES

```
terminal ID      : '^'?('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'_'|'0'..'9')*;
terminal INT returns ecore::EInt: ('0'..'9')+;
```

```
QualifiedName:
    ID ('.' ID)*;

Double returns ecore::EDouble:
    INT ('.' INT)?;
```

```
enum Visibility:
    public="public" | public="+" |
    private="private" | private="-" |
    protected="protected" | protected="#";
```

AST CONSTRUCTION

```
Class :
    (abstract?='abstract')? 'class' name=ID ('extends' super=[Class])?;
```

Example 1: `abstract class A`

First Feature assignment in rule
→Create new instance of type Class
→Assign it to ”current Object”
→Assign value to feature of “current”

Subsequent assignment
→Assign value to feature of “current”

Example 2: `class B extends A`

AST CONSTRUCTION

```
EObject current = null; // result
if (token==‘abstract’) {
  if (current == null) current = new Class();
  current.setAbstract(true);
  advance();
}
advance(‘class’);
if (token ==ID) {
  if (current ==null) current = new Class();
  current.setName(tokenValue);
  advance();
}
...
return current;
```

Clas

(

ass])?;

First F

→Cre

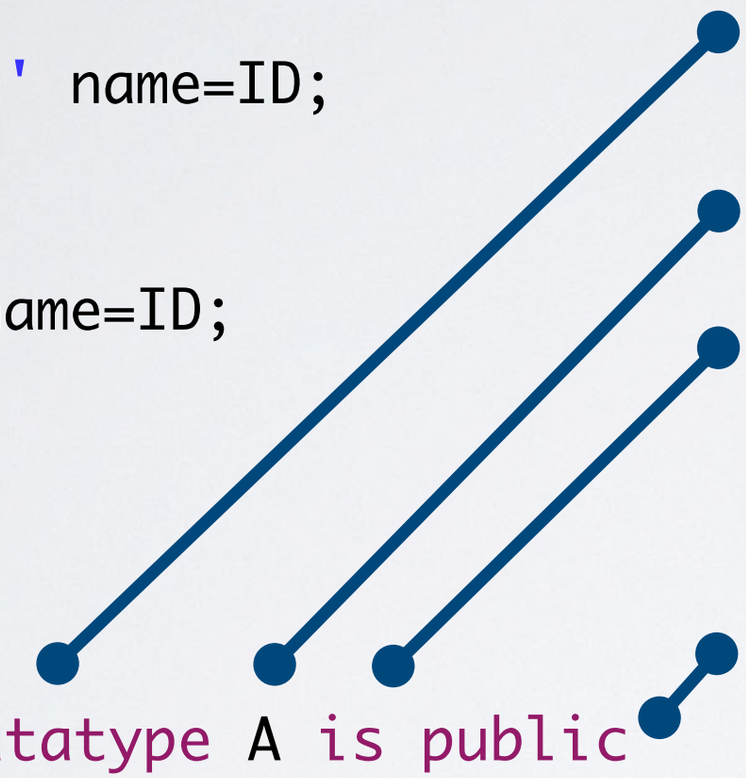
→Assi

→Assi

”

AST CONSTRUCTION II

```
Type :
  (Class | DataType) 'is' visibility=('public' | 'private');

DataType :
  'datatype' name=ID;

Class :
  'class' name=ID;
```



Example `datatype A is public`

Rule "Type" steps into rule "DataType"

→Nested rule creates, initializes and returns DataType

→Parser returns to rule "Type"

→"Current object" of rule "Type" becomes return value of called rule (DataType)

→Subsequent assignments set features of "current object"

- Rule "Type" returns either a Class or a DataType
- In this example only one instance has been created

AST CONSTRUCTION

```
Type
  (C

Data
  'd

Clas
  'c
```

```
EObject current = null; // result
if (token==‘datatype’) {
  current = parseDatatype();
} else { .. }
advance(‘is’);
if (token ==‘public’ || token==‘private’) {
  if (current ==null) current = new Type();
  current.setVisibility(tokenValue);
  advance();
}
...
return current;
```

Exam

- Rul
- In t

UNORDERED GROUPS

```
Employee:
    'firstName' firstName=ID
    ('age' age=INT)?
    'lastName' lastName=ID
    ('yearOfBirth' yearOfBirth=INT)?
;
```

UNORDERED GROUPS

```
Employee:
    'firstName' firstName=ID
    ('age' age=INT)?
    'lastName' lastName=ID
    ('yearOfBirth' yearOfBirth=INT)?
;
```

```
firstName Holger
lastName Schil

yearOfBirth
```

UNORDERED GROUPS

```
Employee:
    ('firstName' firstName=ID
    ('age' age=INT)?
    'lastName' lastName=ID
    ('yearOfBirth' yearOfBirth=INT)?)*
;
```

UNORDERED GROUPS

```
Employee:
    ('firstName' firstName=ID
    ('age' age=INT)?
    'lastName' lastName=ID
    ('yearOfBirth' yearOfBirth=INT)?)*
;
```

```
 5 Employee:
 6     ('firstName' firstName=ID
 7     ('age' age=INT)?
 8     'lastName' lastName=ID
 9     ('yearOfBirth' yearOfBirth=INT)?)*
10 ;
```

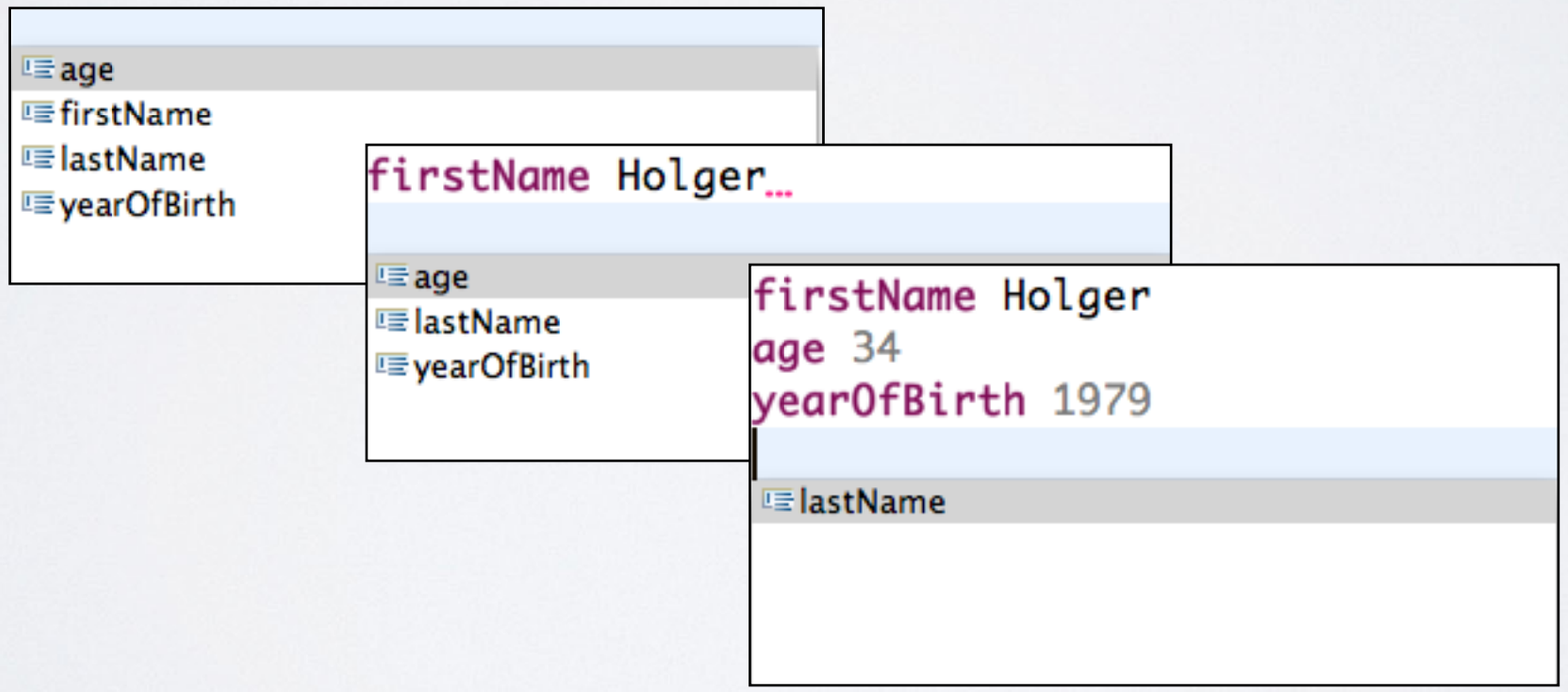
The assigned value of feature 'firstName' will possibly override itself because it is used inside of a loop.

…

UNORDERED GROUPS

```
Employee:
    (,firstName' firstName=ID &
    ('age' age=INT)? &
    'lastName' lastName=ID &
    ('yearOfBirth' yearOfBirth=INT)?)
;
```

Each element can only appear once but in any order



Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- **EMF / Ecore integration**
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

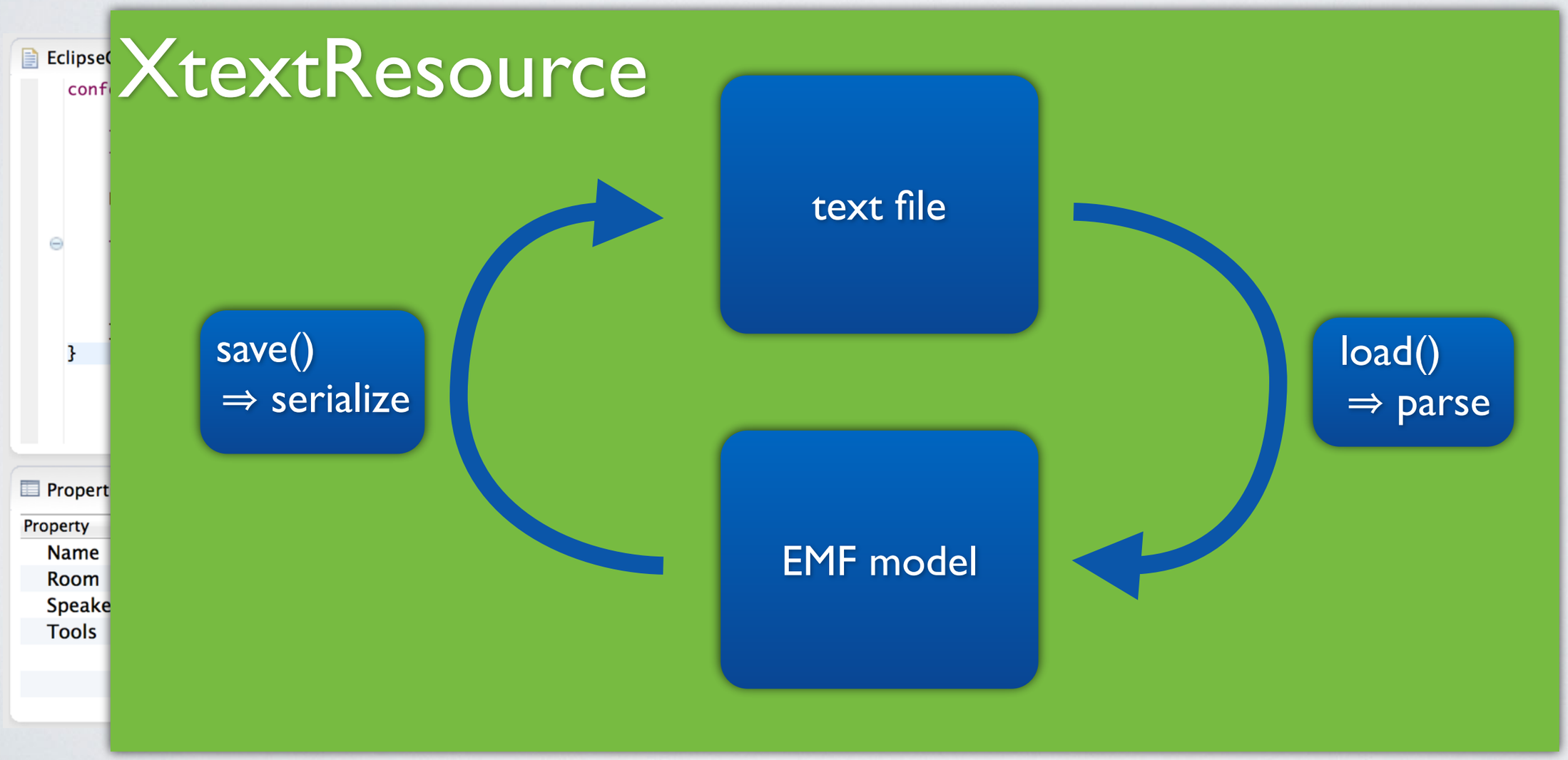
Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

EMF INTEGRATION

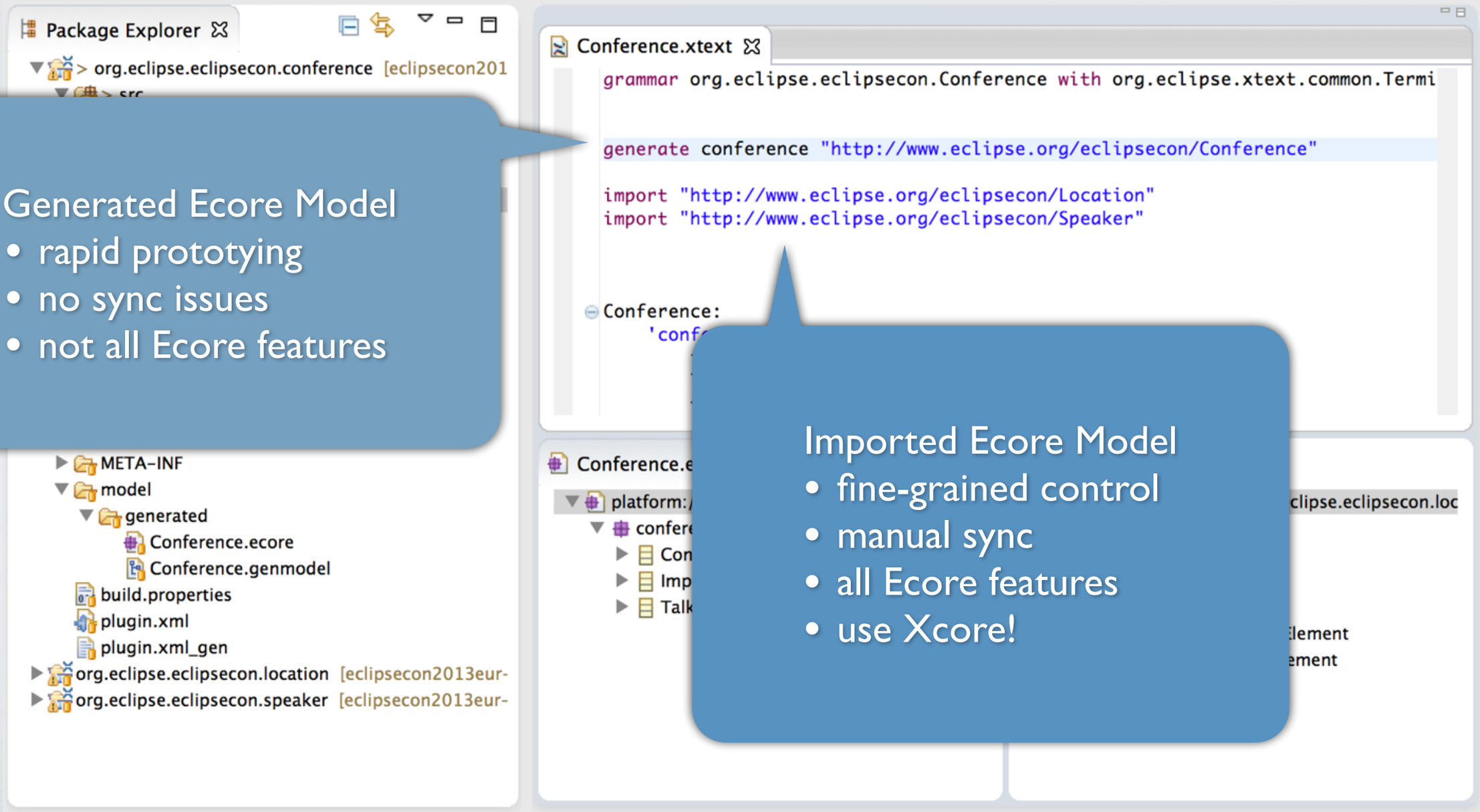
EMF ECLIPSE MODELING ECOSYSTEM

your text files are EMF models!



EMF ECLIPSE MODELING ECOSYSTEM

generate vs. import Ecore model



Generated Ecore Model
- rapid prototying
- no sync issues
- not all Ecore features

Imported Ecore Model
- fine-grained control
- manual sync
- all Ecore features
- use Xcore!

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- **Index**
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

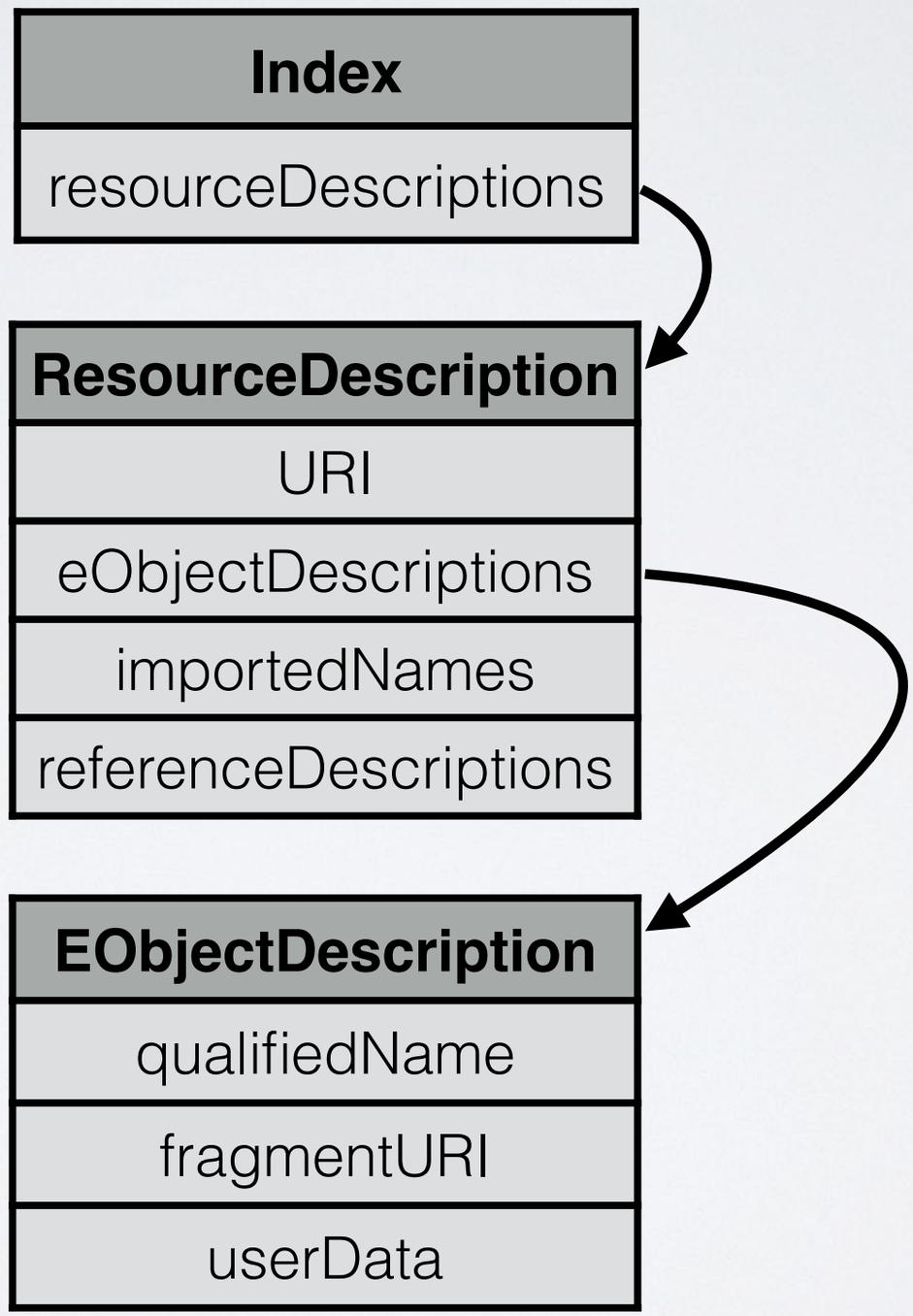
Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

INDEX?

- enables filename-agnostic resolution of names
- enables validation of cross-references without loading targets
- enables computation of affected files during incremental build
- reverse-reference lookup

INDEX STRUCTURE



INDEX POPULATION

resource change

<triggers>

incremental Build

```
for(resource : changedOrAffected) {
	val description = new ResourceDescription(resource.uri)
	for(eObject : resource.eAllContents) {
		val name = qualifiedNameProvider.get(eObject)
		if(name != null) {
			val fragment = resource.getFragmentURI(eObject)
			description.add(new EObjectDescription(name, fragment))
		}
	}
	index.add(description)
}
for(resource : changedOrAffected) {
	EcoreUtil.resolveAll(resource)
	validator.validate(resource)
}
```

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- **UI customizations**
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

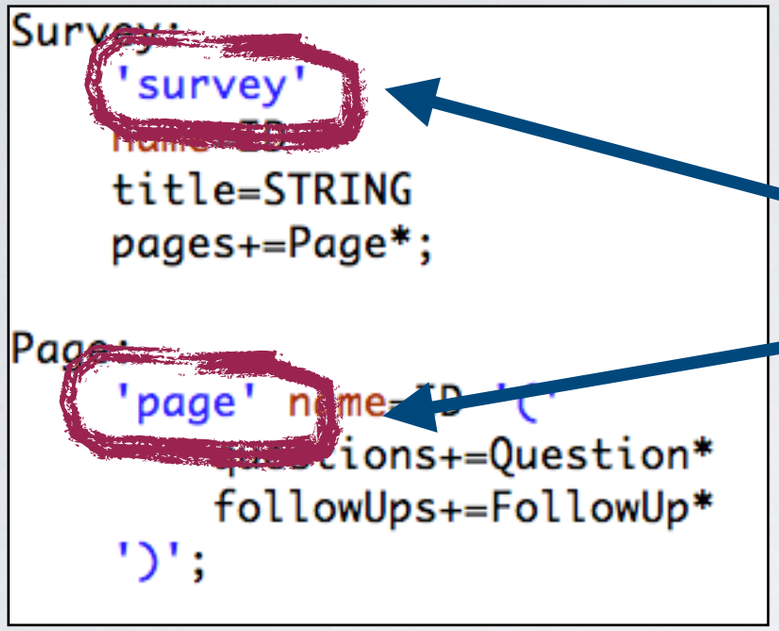
9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

CODE COMPLETION

From Grammar

```
Survey:
    'survey'
    name=ID
    title=STRING
    pages+=Page*;

Page:
    'page' name=ID '('
        questions+=Question*
        followUps+=FollowUp*
    ')';
```

From Grammar



Keywords

From Grammar

```
Survey:
    'survey'
    name=ID
    title=STRING
    pages+=Page*;

Page:
    'page' name=ID '('
        questions+=Question*
        followUps+=FollowUp*
    ')';
```

Keywords

From Scope

```
FollowUp:
    guard=Guard? '->' next=[Page];

Guard:
    'if'
    question=[ChoiceQuestion|QualifiedName]
    '='
    answer=[Choice];
```

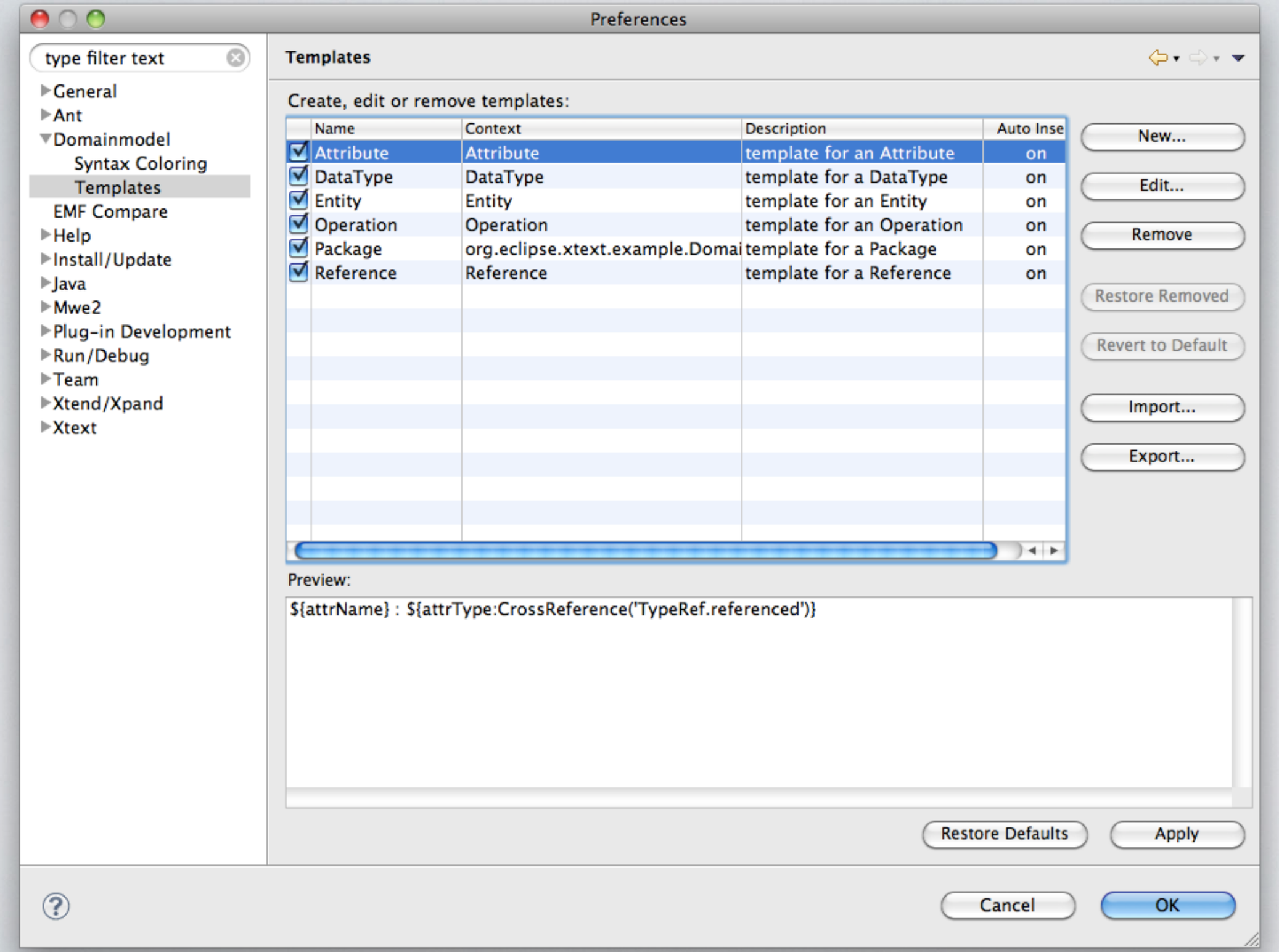
```
class SurveyProposalProvider extends AbstractSurveyProposalProvider {

    co
}
```

completeAssignment(Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractJavaBasedContentProposalProvider
completeChoiceQuestion_Choices(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeChoiceQuestion_Name(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeChoiceQuestion_Single(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeChoiceQuestion_Text(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeChoice_Freetext(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeChoice_Name(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeChoice_Text(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeFollowUp_Guard(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeFollowUp_Next(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeFreeTextQuestion_Name(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeFreeTextQuestion_Text(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeGuard_Answer(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeGuard_Question(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeKeyword(Keyword, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractJavaBasedContentProposalProvider
completePage_FollowUps(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completePage_Name(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completePage_Questions(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeRuleCall(RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractJavaBasedContentProposalProvider
completeSurvey_Name(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeSurvey_Pages(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
completeSurvey_Title(EObject, Assignment, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_ANY_OTHER(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider
complete_Choice(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_ChoiceQuestion(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_FollowUp(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_FreeTextQuestion(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_Guard(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_ID(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider
complete_INT(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider
complete_ML_COMMENT(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider
complete_Page(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_QualifiedName(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_Question(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_SL_COMMENT(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider
complete_STRING(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider
complete_Survey(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from AbstractSurveyProposalProvider
complete_WS(EObject, RuleCall, ContentAssistContext, ICompletionProposalAcceptor) – Override method from TerminalsProposalProvider

blems
inated>
[main]
[main]

TEMPLATE PROPOSALS

TEMPLATES


Preferences
type filter text
General
Ant
Domainmodel
Syntax Coloring
Templates
EMF Compare
Help
Install/Update
Java
Mwe2
Plug-in Development
Run/Debug
Team
Xtend/Xpand
Xtext
Templates
Create, edit or remove templates:
Name
Context
Description
Auto Inse
Attribute
Attribute
template for an Attribute
on
DataType
DataType
template for a DataType
on
Entity
Entity
template for an Entity
on
Operation
Operation
template for an Operation
on
Package
org.eclipse.xtext.example.Domai
template for a Package
on
Reference
Reference
template for a Reference
on
New...
Edit...
Remove
Restore Removed
Revert to Default
Import...
Export...
Preview:
${attrName} : ${attrType:CrossReference('TypeRef.referenced')}
Restore Defaults
Apply
Cancel
OK

TEMPLATE PROPOSALS

org.eclipse.jface.text.templates.GlobalTemplateVariables

${time}

${user}

${date}

${cursor}

CROSSREFERENCES

org.eclipse.xtext.ui.editor.templates.CrossReferenceTemplateVariableResolver

```
${ref:CrossReference(MyType.ref)}
```

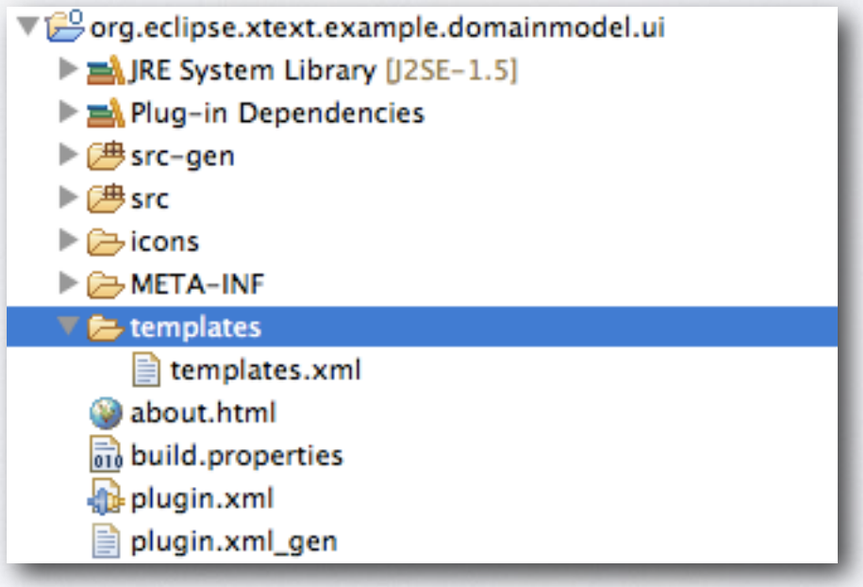
ENUMERATION

org.eclipse.xtext.ui.editor.templates.EnumTemplateVariableResolver

```
${visibility:Enum(Visibility)}
```

EXPORT TEMPLATES

```
<templates>
    <template name="Package" description="template for a Package"
      id="org.eclipse.xtext.example.Domainmodel.Package"
      context="org.eclipse.xtext.example.Domainmodel.Package"
      enabled="true">package ${name} {
        ${cursor}
      }
</template>
```

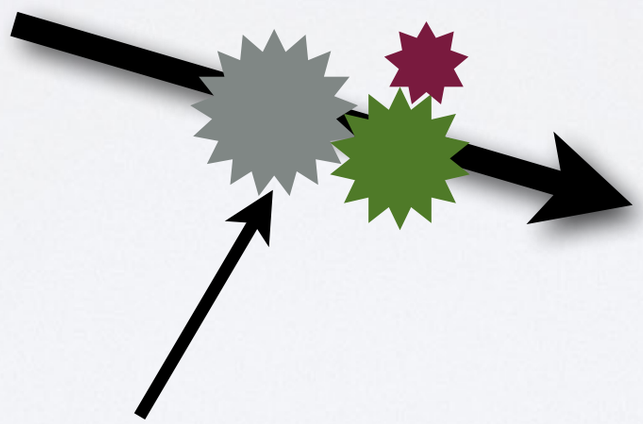


FORMATTING

FORMATTING

Pretty Printing & Code Beautification

```
     import
<Literature.entities> entity
City {Int   id;       String
name;}
  entity Store { Int
id;      String       name;
Int
city
   references
City.
id;   }
```



§ 1. At … do …
§ 2. At … do …
§ 3. At … do …
§ 4. At … do …
…

```
import <Literature.entities>

entity City {
     Int id;
     String name;
}

entity Store {
     Int id;
     String name;
     Int city references
     City.id;
}
```

FORMATTING

```
@Inject extension DomainmodelGrammarAccess

override protected void configureFormatting(FormattingConfig c) {
    c.autoLinewrap = 120

    c.setLinewrap(1, 2, 3).around(abstractElementRule)
    c.setLinewrap(1, 2, 3).around(packageDeclarationRule)
    c.setLinewrap(1, 1, 2).around(featureRule)
    c.setNoSpace().before(typeRefAccess.multiAsteriskKeyword_1_0)

    val pairs = findKeywordPairs("{", "}")
    for (pair : pairs) {
        c.setIndentation(pair.first, pair.second)
    }

    c.setLinewrap(0, 1, 2).before(SL_COMMENTRule)
    c.setLinewrap(0, 1, 2).before(ML_COMMENTRule)
    c.setLinewrap(0, 1, 1).after(ML_COMMENTRule)
}
```

QUICK FIX

Validation

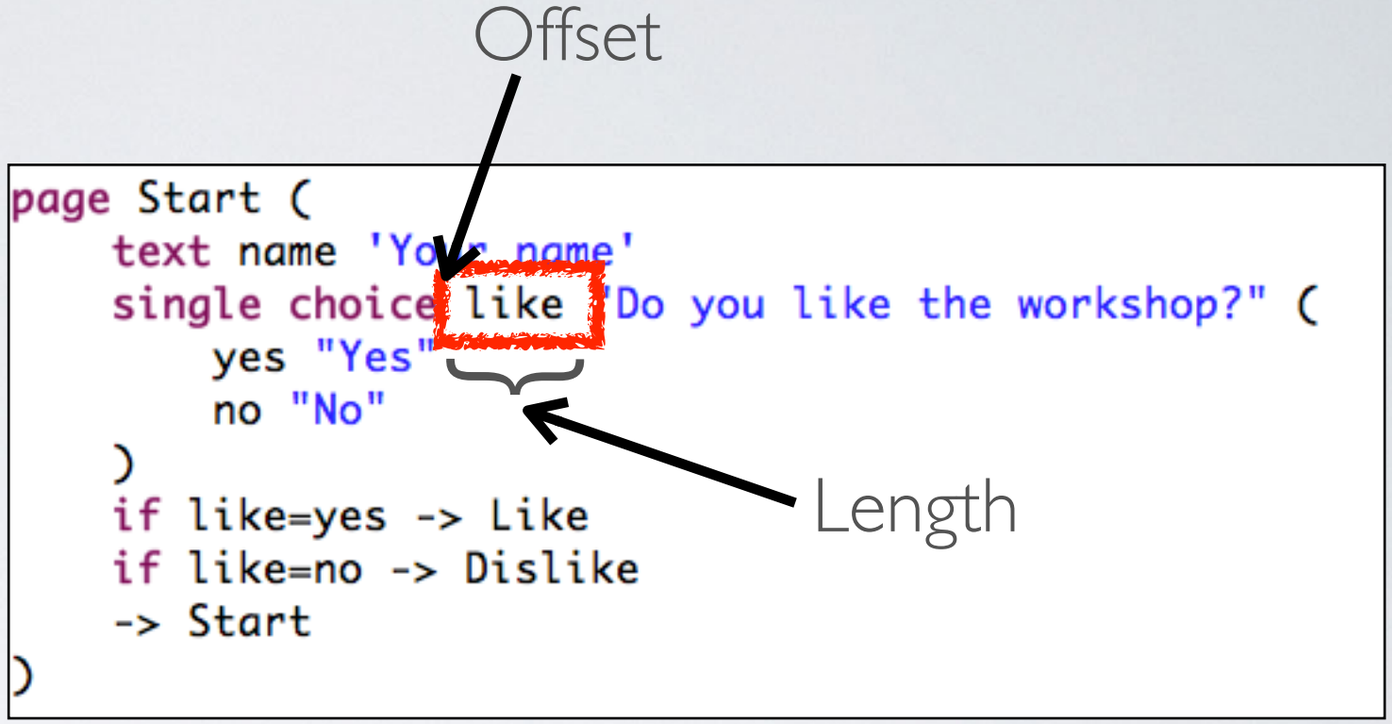
```
public static final String INVALID_NAME = "org.xtext.quickfix.InvalidTypeName";

@Check
def void checkTypeNameStartsWithCapital(City city) {
    if (city.getName() == null || city.getName().length() == 0) return;
    if (!Character.isUpperCase(city.getName().charAt(0))) {
        warning("Name should start with a capital letter.",
                ExampleQuickfixPackage.Literals.CITY__NAME, INVALID_NAME,
                city.getName());

    }
}
```

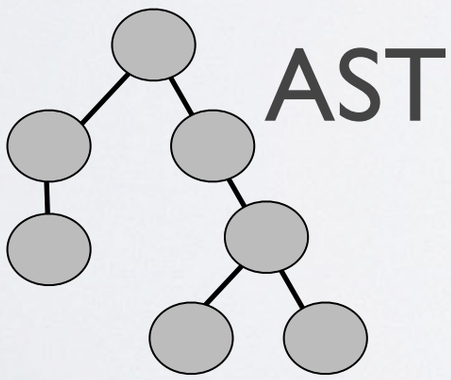
Fix

```
@Fix(ExampleQuickfixValidator.INVALID_NAME)
def void capitalizeName(Issue issue, IssueResolutionAcceptor acceptor) {
    acceptor.accept(issue, "Capitalize name",
        "Capitalize the name \"" // retrieve the 'user data' from the validation warning
        + issue.getData().head + "\".", // icon to display (in the icons folder)
        "upcase.png",
        [context |
            val xtextDocument = context.getXtextDocument();
            val firstLetter = xtextDocument.get(issue.getOffset() + 1, 1);
            xtextDocument.replace(issue.getOffset() + 1, 1, firstLetter.toUpperCase());
        ]);
}
```

Different types of quick fixes

On a textual level





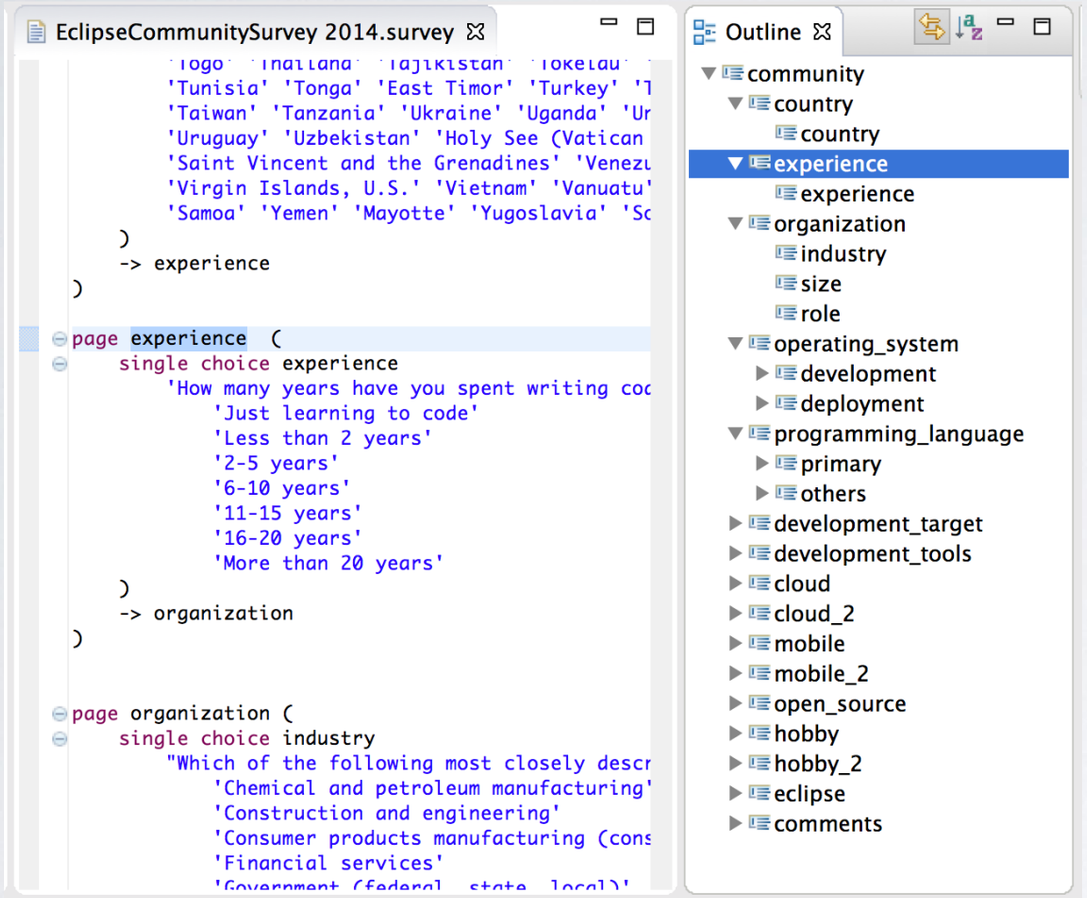
On AST level

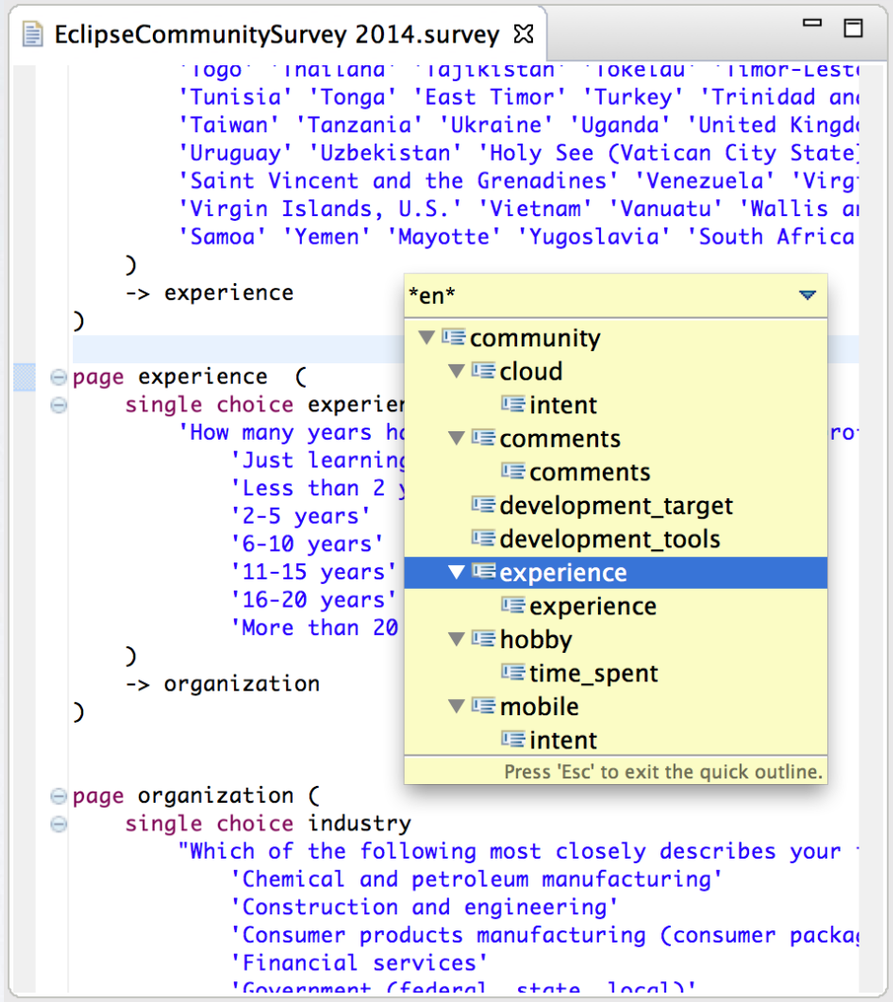
SYNTAX HIGHLIGHTING

Demo

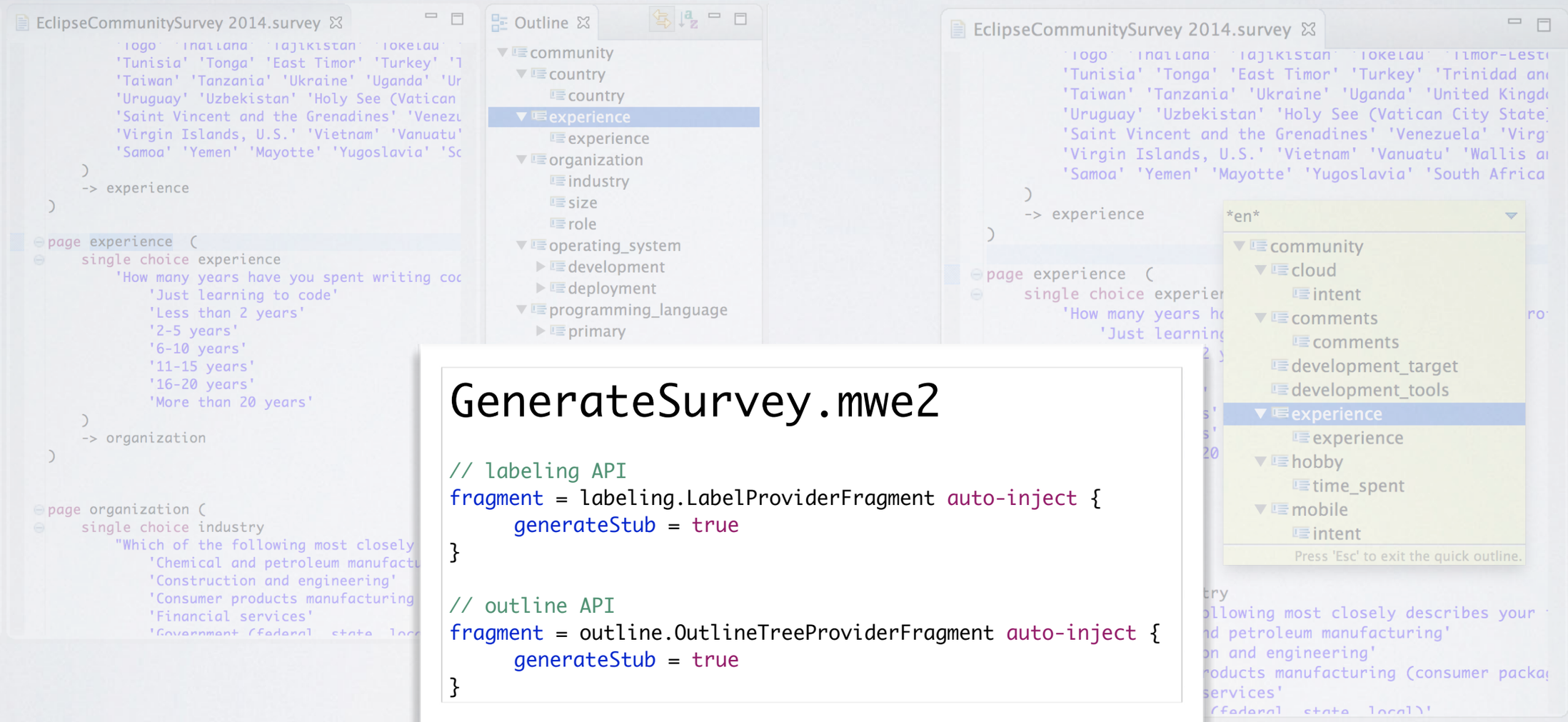
OUTLINE VIEW

OUTLINE VIEW





OUTLINE VIEW



GenerateSurvey.mwe2

```
// labeling API
fragment = labeling.LabelProviderFragment auto-inject {
      generateStub = true
}

// outline API
fragment = outline.OutlineTreeProviderFragment auto-inject {
      generateStub = true
}
```

```
org.eclipse.xtext.tutorial.survey.ui.outline.SurveyOutlineTreeProvider
org.eclipse.xtext.tutorial.survey.ui.labeling.SurveyLabelProvider
```

Outline

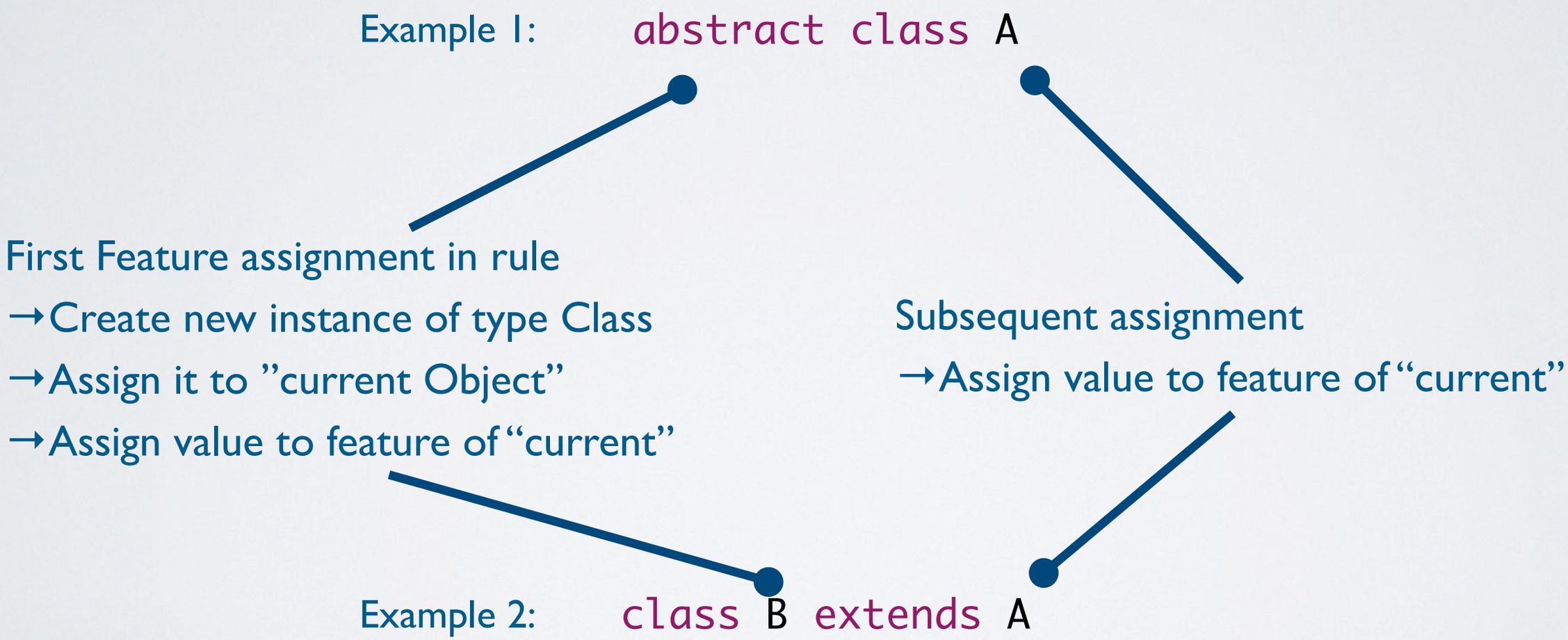
- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- **Assigned Actions**
- Testing
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

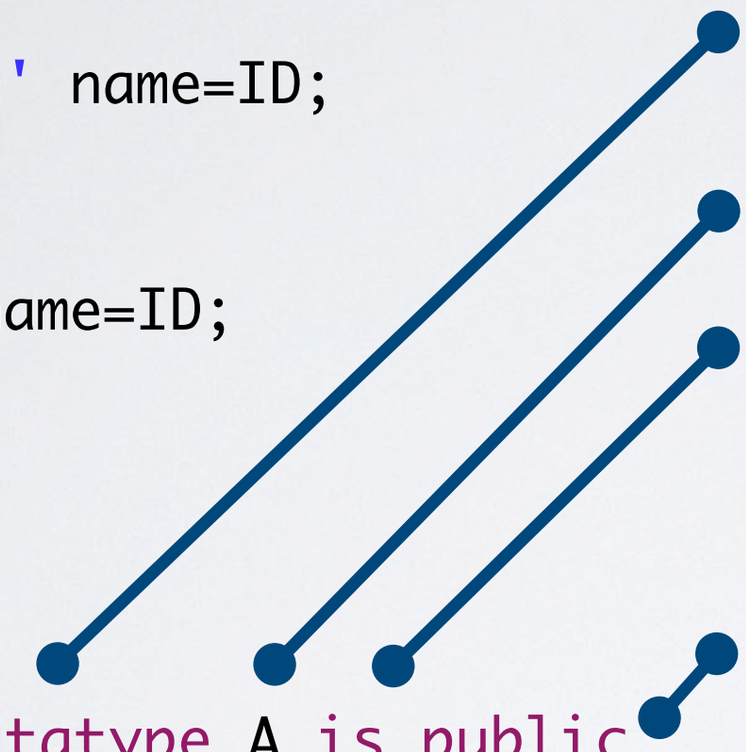
RECAP: AST CONSTRUCTION

```
Class :
    (abstract?='abstract')? 'class' name=ID ('extends' super=[Class])?;
```



RECAP: AST CONSTRUCTION II

```
Type :
  (Class | DataType) 'is' visibility=('public' | 'private');

DataType :
  'datatype' name=ID;

Class :
  'class' name=ID;
```



Example `datatype A is public`

Rule "Type" steps into rule "DataType"

→Nested rule creates, initializes and returns DataType

→Parser returns to rule "Type"

→"Current object" of rule "Type" becomes return value of called rule (DataType)

→Subsequent assignments set features of "current object"

- Rule "Type" returns either a Class or a DataType
- In this example only one instance has been created

UNASSIGNED ACTIONS

```
Questionnaire:
  (answers+=Answer)*;

Answer:
  {Answer} 'no' | yes?='yes';
```

Assignment triggers object creation

Object creation triggered by action

Example: yes no

- Without the action a no would never create an AST element

EXPRESSIONS

```
Expression :
  Expression '+' Expression |
  '(' Expression ')' |
  Number;

Number :
  value = INT;
```

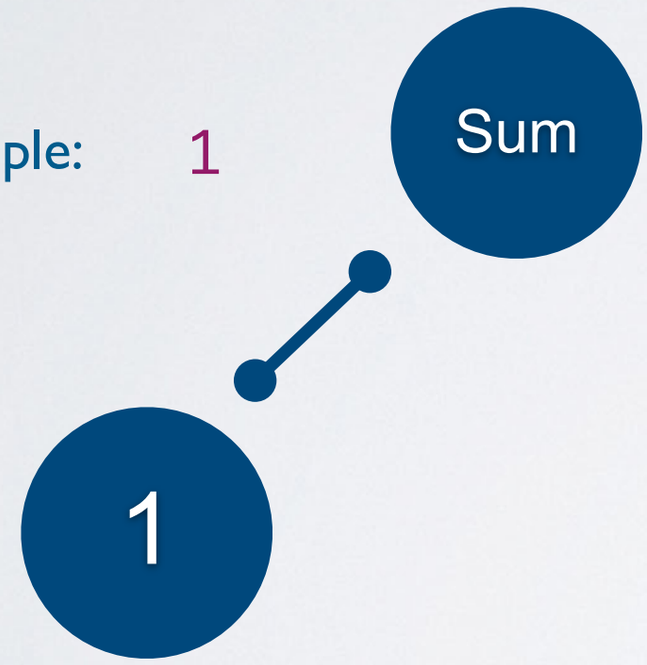
LEFT-RECURSION !

EXPRESSIONS

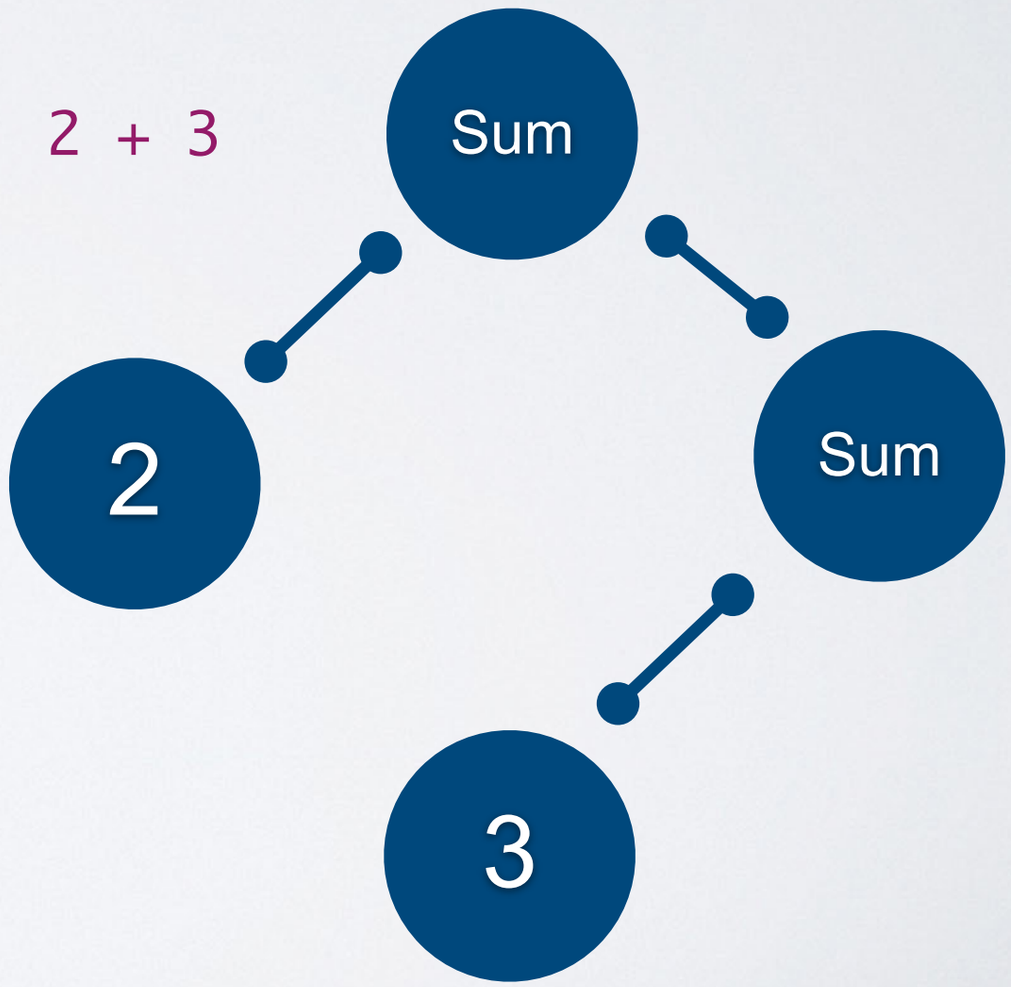
D-

```
Sum :
  left=Number ('+' right=Sum)?;

Number :
  value = INT;
```

Example: 1



Example: 2 + 3

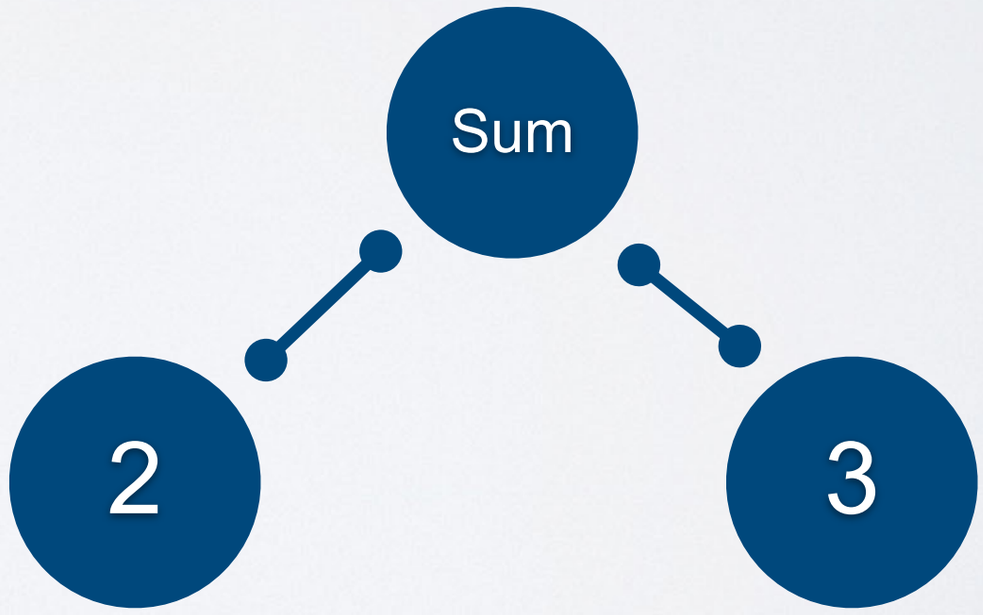


EXPRESSIONS

```
Addition returns Expression:
  Number ({Sum.left = current} '+' right=Number)*;

Number :
  value = INT;
```

Example: 1

Example: 2 + 3



Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- **Testing**
- Xbase
- Problem Solving
- Outlook

Lunch Coffee

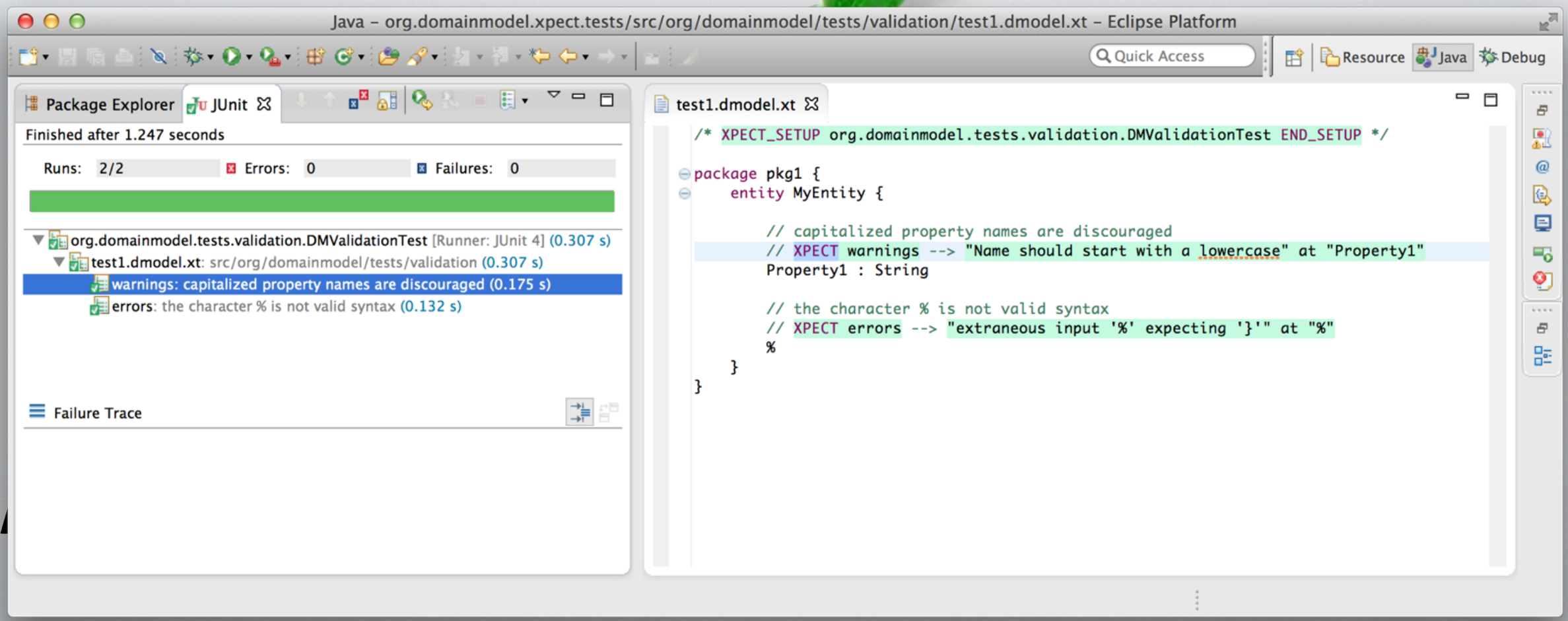
9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

AUTOMATED TESTING

Xtext: JUnit utils:

XtextRunner, ParseHelper, ValidatorTester, IResourcesSetupUtil, JavaProjectSetupUtil...

GitHub: Xpect:



Xpect: reusable Tests as Library for...

- Parser and AST (demo only, no library).
- Code generators
- Validation
- Linking
- Scoping
- ResourceDescriptions
- JvmModelInferrer

DEMO!

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- **Xbase**
- Problem Solving
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

XBASE

Enable expressions in your language

XBASE

Enable expressions in your language

Translate your DSL concepts to Java concepts

XBASE

Enable expressions in your language

Translate your DSL concepts to Java concepts

Bring your expressions into context

XBASE

Enable expressions in your language

Translate your DSL concepts to Java concepts

Bring your expressions into context

Get seamless IDE integration

XBASE

DEMO

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- **Problem Solving**
- Outlook

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

HOW TO FIND A GUICE BINDING CANDIDATE I

- Open the (Ui)Module of your language
- Find a method containing the interface‘s name in its name
  - 2x CTRL-O to include superclasses
- If there‘s nothing, open the interface and look for a @ImplementedBy annotation

HOW TO FIND A GUICE BINDING CANDIDATE 2

- Eclipse Navigation Features are the Swiss Army Knife
- Open Type Dialog `*.xtext.*.*<Aspect>`
- Type Hierarchy to Find Implementors
- Find Usages
- Call Hierarchy

HOW TO TWEAK SOME FEATURE I

- Is it documented in
  - Eclipse help,
  - the Xtext forum / newsgroup,
  - one of our blogs?
- If so:
  - Read these docs!

HOW TO TWEAK SOME FEATURE 2

- Is it available in one of the examples?
- If so: Monkey see - Monkey do. Look for
  - unusual generator fragments
  - custom bindings
  - extensions in plugin.xml

HOW TO TWEAK SOME FEATURE 3

- Is there an implementation of IGeneratorFragment ?
- If so:
  - Use the fragment in your workflow
  - Investigate its properties using code assist or looking at the code
  - Watch out for new stub classes in the src folders

HOW TO TWEAK SOME FEATURE 4

- Look for Xtext classes/packages whose names suggest to deal with the feature.
- If there is one:
  - Try to find the hook (interface binding), e.g. by investigating code, debugging, etc.
  - Rather inherit from existing code than write from scratch
  - The JavaDocs should guide you

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Cross-References and Scoping
- Advanced Grammar
- EMF / Ecore integration
- Index
- UI customizations
- Assigned Actions
- Testing
- Xbase
- Problem Solving
- **Outlook**

Lunch Coffee

9:30 - 10:45 | 11:00 - 12:00 | 13:00 - 14:30 | 15:00 - 16:30

SESSIONS YOU SHOULD ATTEND ON TUESDAY

Xtext Grammar Language
Jan Köhnlein

Scoping, Linking and Indexing
Moritz Eysholdt

Understanding Dependency Injection in Xtext
Stefan Oehme

Xbase - The Complete Guide
Sebastian Zarnekow

Performance Optimization
Sebastian Zarnekow

Testing with Xpect
Moritz Eysholdt

SESSIONS YOU SHOULD ATTEND ON WEDNESDAY

What's in a resource?

Sebastian Zarnekow

Code Generation with Xtend

Jan Köhnlein

Implementing Interpreters

Anton Kosyakov

Advanced Parsing and Lexing

Sebastian Zarnekow

Thank You