

Installation

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

Xtext for Beginners

Anton Kosyakov, Moritz Eysholdt
itemis

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Introduce Cross-References
- Outlook

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Introduce Cross-References
- Outlook

Eclipse Community Survey X

localhost:8080/programming_language.html

Eclipse Community Survey 2014 Evaluate

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

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

Demo

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

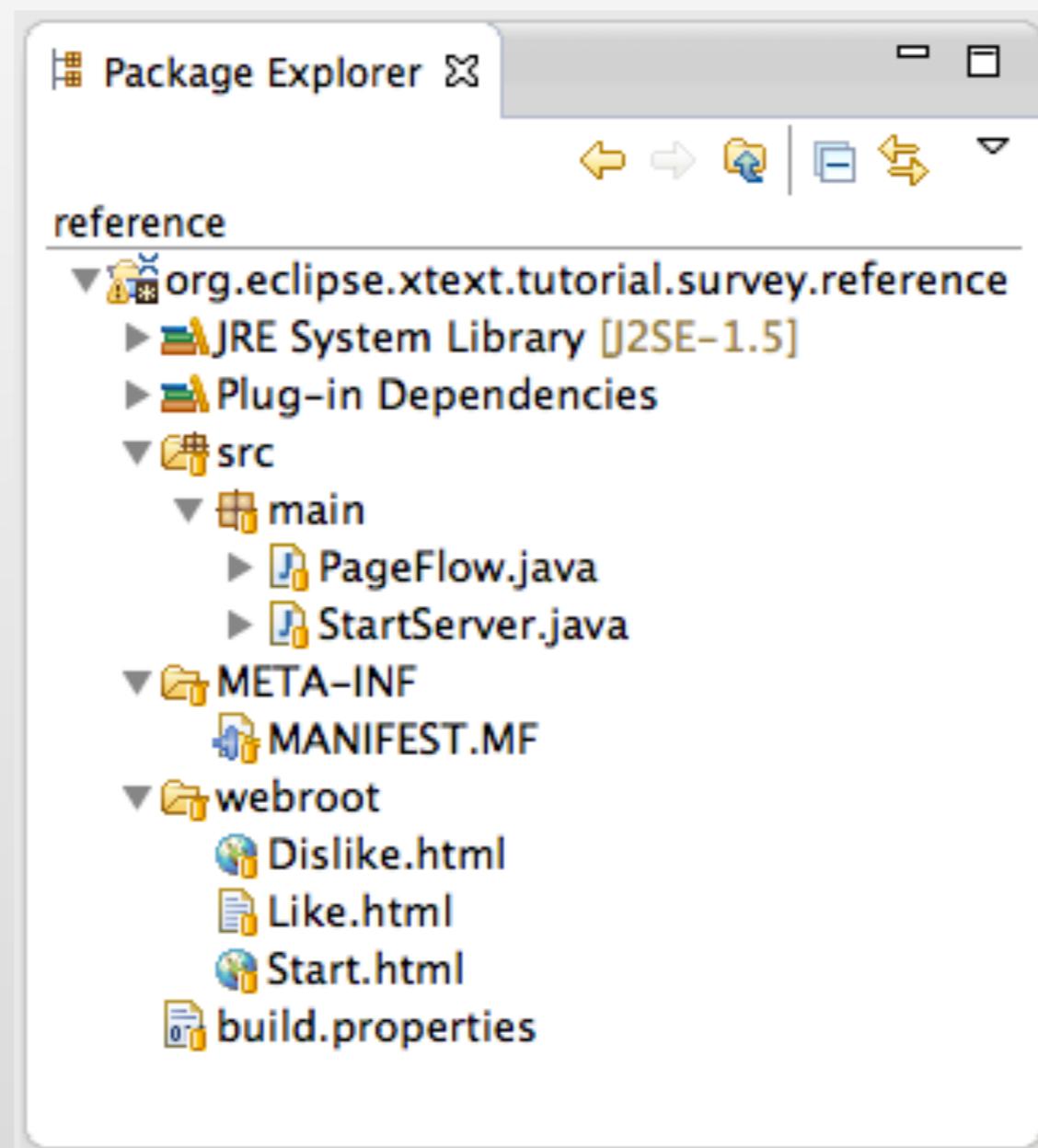
Example: Surveys

- A Web-Based App for Surveys
- Multiple Pages
- Multiple Questions per Page
- Different Types of Questions

Architecture

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

Reference Project



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

Page

ChoiceQuestion

Choice

Choice

Choice

Page

ChoiceQuestion

Choice

Choice

Choice

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

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

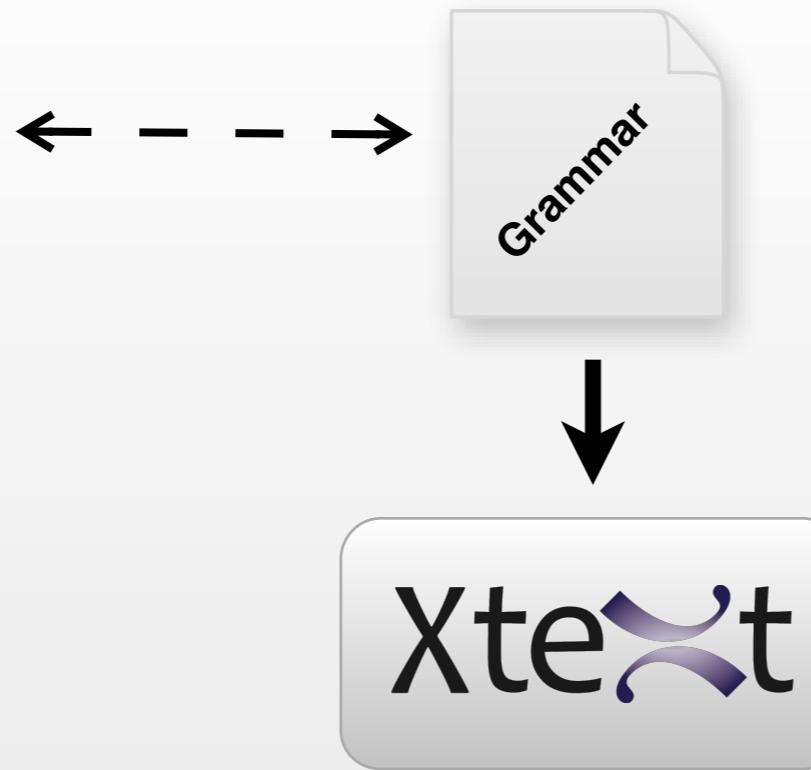
Advantages

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

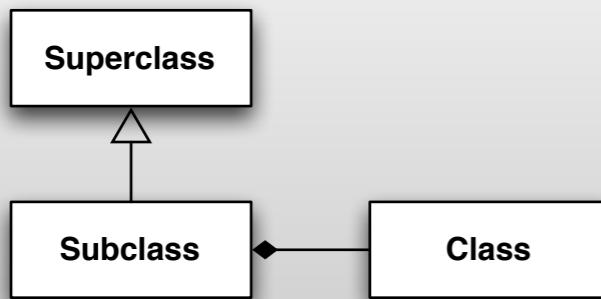
Outline

- Example Application
- **Build your DSL with Xtext**
- Generate Code with Xtend
- Validate Models
- Introduce Cross-References
- Outlook

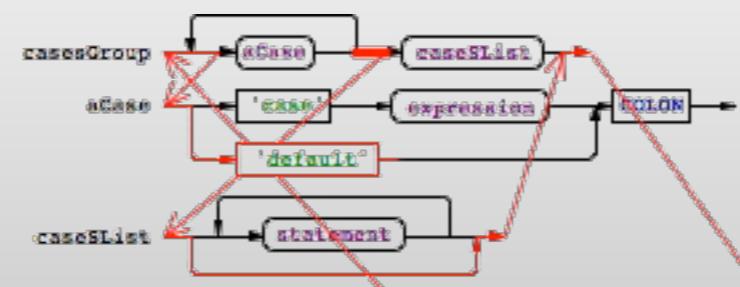
Reference Model



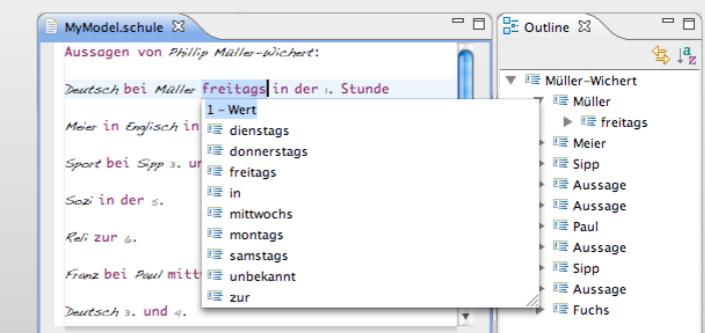
Xtext Runtime



Ecore Package



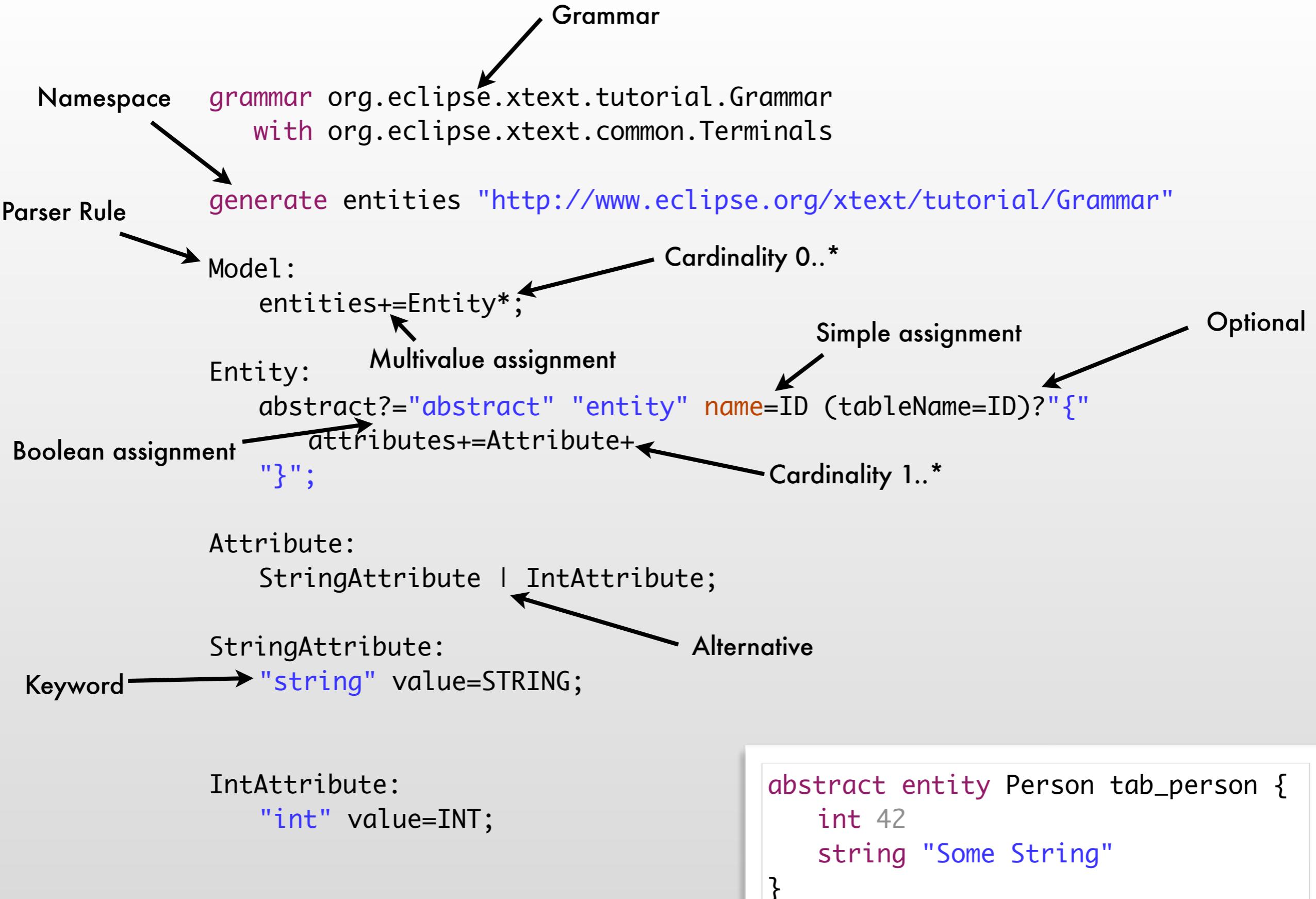
Parser



Editor

Exercise 0

Create a New Xtext Project



```

abstract entity Person tab_person {
    int 42
    string "Some String"
}

```

Exercise 1

Write a Grammar for an Example File

Outline

- Example Application
- Build your DSL with Xtext
- **Generate Code with Xtend**
- Validate Models
- Introduce Cross-References
- Outlook



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.

For decisions there is the IF statement

```
«IF elements.empty»  
    no elements.
```

```
«ELSE»
```

You can also iterate a collection with FOR:

```
«FOR element : elements.indexed»  
    Element «element.key + 1»: «element.value»  
«ENDFOR»  
«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 2

Implement a Code Generator

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- **Validate Models**
- Introduce Cross-References
- Outlook

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 3

Implement a Custom Validation Rule

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Introduce Cross-References
- Outlook

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]  
)';
```

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 '...'
    )
...
)
```

question=[Question QualifiedName]		
"page Start"	File	Global
name	Start.name	Other.id
like	Start.like	
	Like.particular	

answer=[Choice ID]
yes
no

default impl

custom impl

Exercise 4

Customize Linking and Content Assist via Scoping

Outline

- Example Application
- Build your DSL with Xtext
- Generate Code with Xtend
- Validate Models
- Introduce Cross-references
- Outlook

Content Assist

Quick Fixes

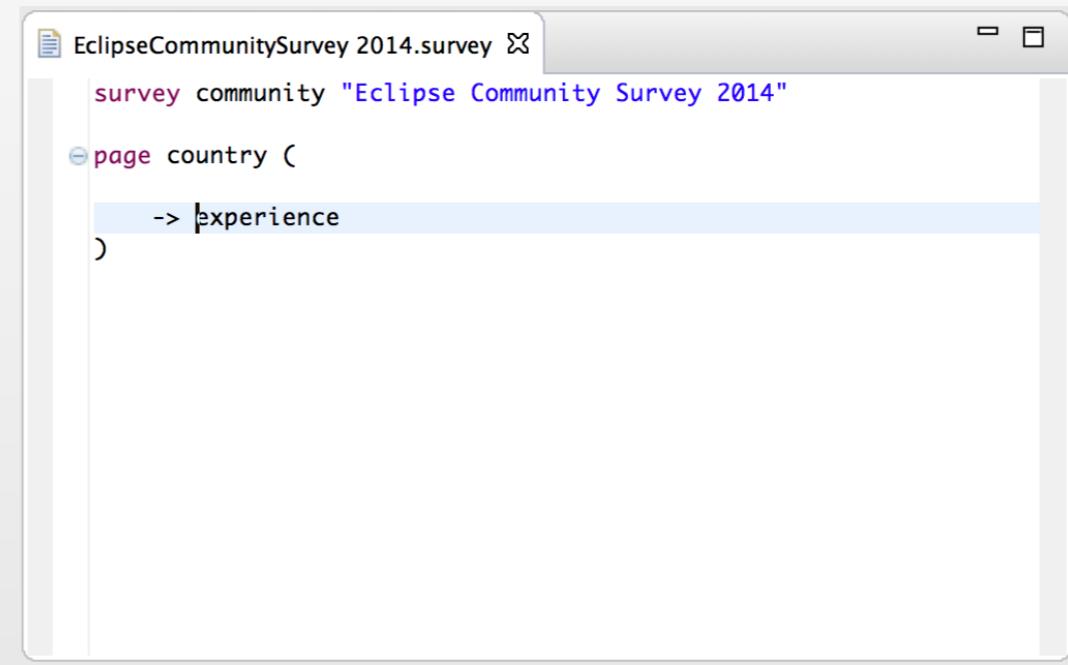
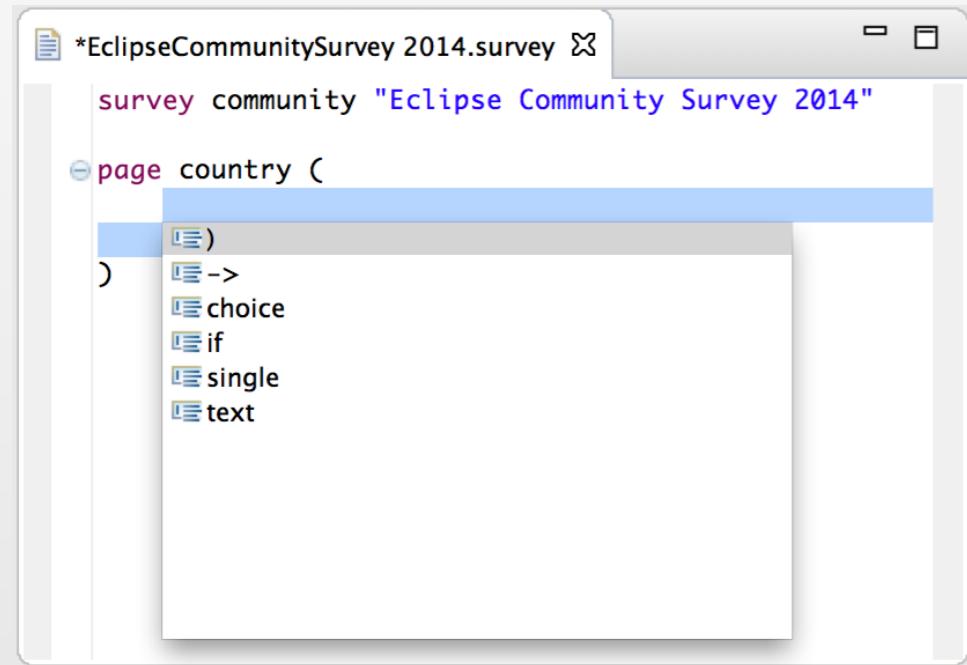
Syntax
Highlighting

Outline

Template
Proposals

Formatting

etc.



By default, it shows

- keyword (based on grammar)
- names of cross references (based on grammar and scoping)
- template proposals (based on context)

Code Pointer:

org.eclipse.xtext.tutorial.survey.ui.contentassist.SurveyProposalProvider

Content
Assist

Quick
Fixes

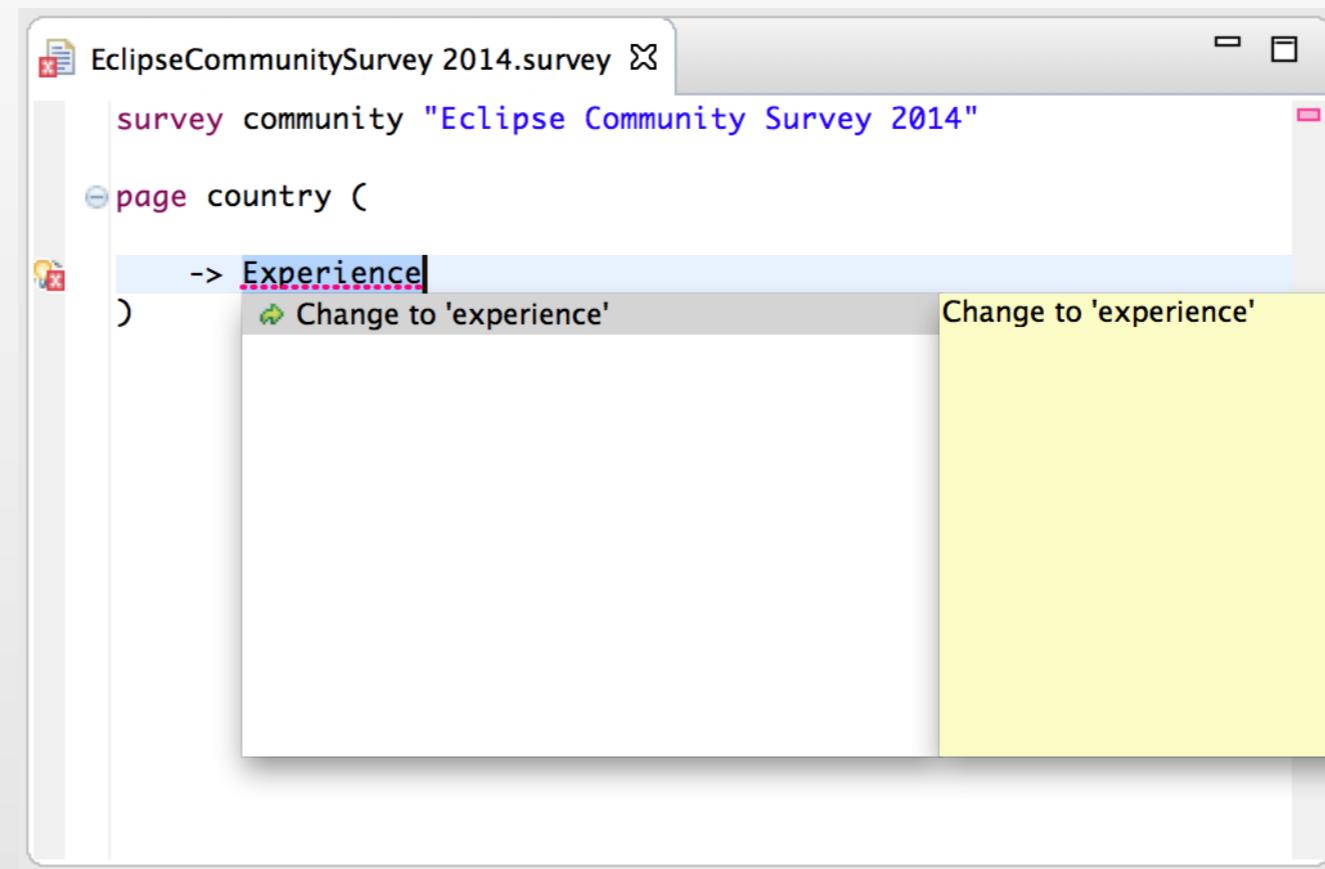
Syntax
Highlighting

Outline

Template
Proposals

Formatting

etc.



Default:

- mostly nothing ;)
- for unresolved cross references, guess reference name

Code Pointer:

org.eclipse.xtext.tutorial.survey.ui.quickfix.SurveyQuickfixProvider
org.eclipse.xtext.ui.editor.model.edit.IModification
org.eclipse.xtext.ui.editor.model.edit.ISemanticModification

Content Assist

Quick Fixes

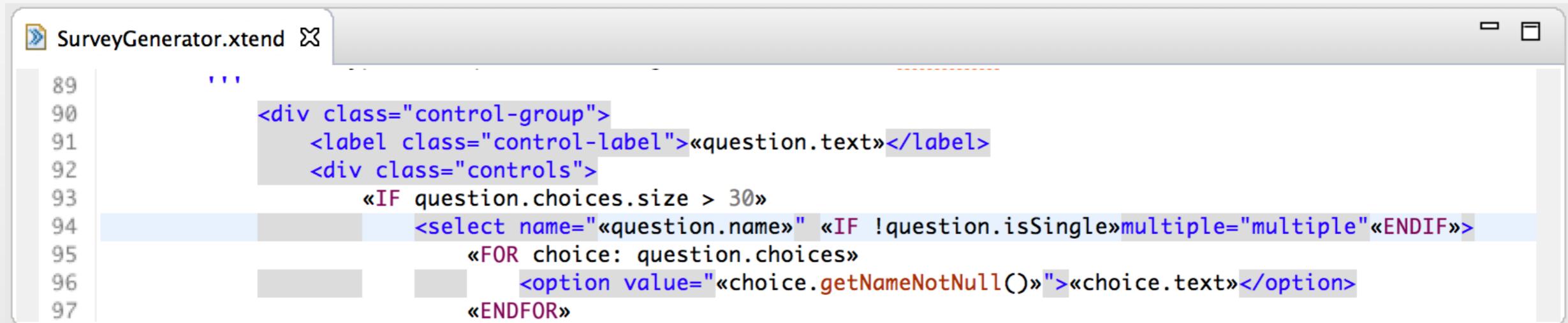
Syntax Highlighting

Outline

Template Proposals

Formatting

etc.



The screenshot shows an IDE editor window titled "SurveyGenerator.xtend". The code is written in Xtend, a domain-specific language for Eclipse. The code uses template strings and annotations like @UI and @Model. Syntax highlighting is applied to different parts of the code, such as XML tags in blue, Java-like code in grey, and template expressions in red. The editor interface includes a toolbar at the top and a status bar at the bottom.

```
89 ...
90     <div class="control-group">
91         <label class="control-label">«question.text»</label>
92         <div class="controls">
93             «IF question.choices.size > 30»
94             <select name="«question.name»" «IF !question.isSingle»multiple="multiple"«ENDIF»>
95             «FOR choice: question.choices»
96                 <option value="«choice.getNameNotNull()»">«choice.text»</option>
97             «ENDFOR»
```

Default: Lexer-based.

Code Pointers:

org.eclipse.xtext.ui.codetemplates.ui.highlighting.SemanticHighlighter
org.eclipse.xtext.ui.editor.syntaxcoloring.DefaultHighlightingConfiguration

Content
Assist

Quick
Fixes

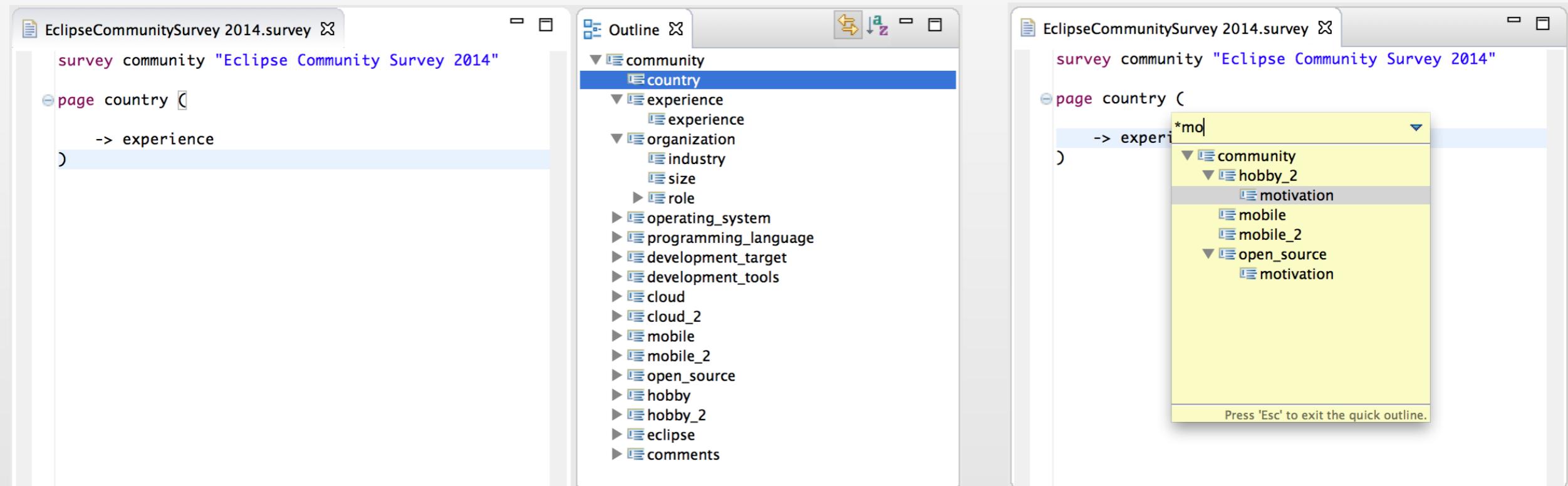
Syntax
Highlighting

Outline

Template
Proposals

Formatting

etc.



Default: Semantic Model

Code Pointer:

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

Content Assist

Quick Fixes

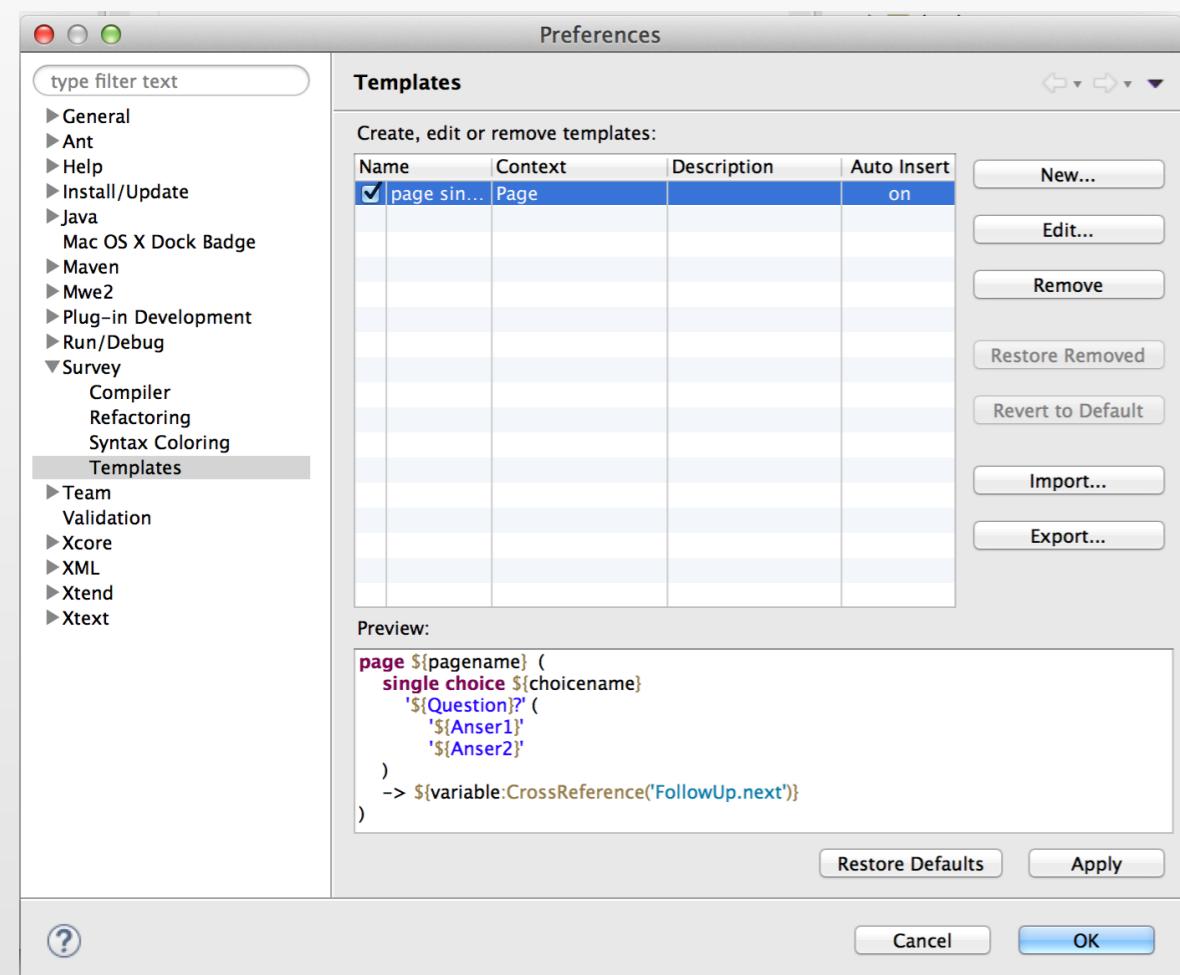
Syntax Highlighting

Outline

Template Proposals

Formatting

etc.



Default: no templates available

User-Customizable!

The screenshot shows two Eclipse Xtext editor windows. The top window is titled 'EclipseCommunitySurvey 2014.survey' and contains the following code:

```
survey community "Eclipse Community Survey 2014"

page
page pagename (
```

A tooltip on the right side of the screen provides details about the 'page' template:

```
page pagename (
    single choice choicename
    'Question?' (
        'Answer1'
        'Answer2'
    )
    -> experience
)
```

The bottom window is titled '*EclipseCommunitySurvey 2014.survey' and contains the same code, with the 'experience' part of the template expanded:

```
survey community "Eclipse Community Survey 2014"

page pagename (
    single choice choicename
    'Question?' (
        'Answer1'
        'Answer2'
    )
    -> experience
)
```

Content Assist

Quick Fixes

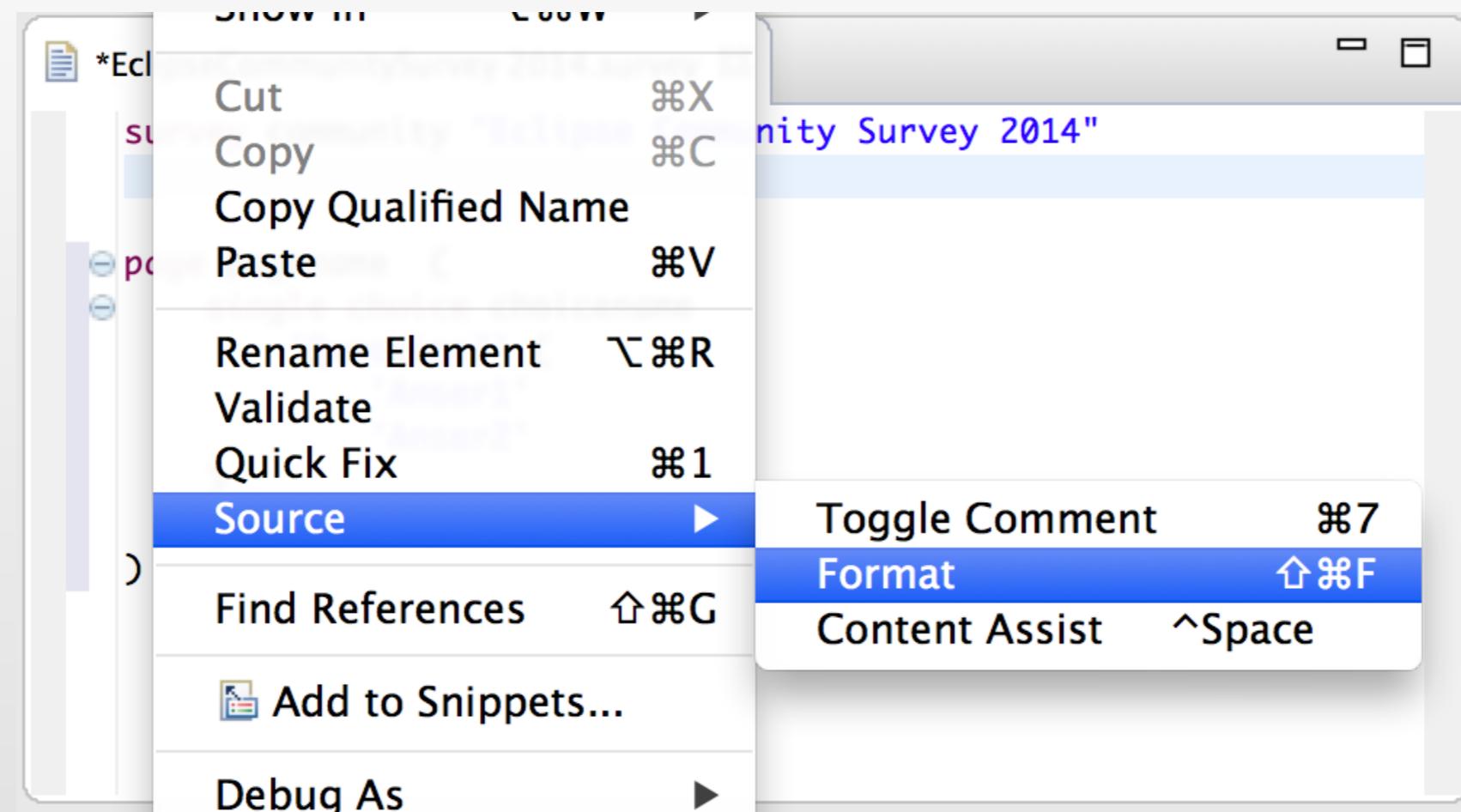
Syntax Highlighting

Outline

Template Proposals

Formatting

etc.



Default: insert single whitespace between tokens

Code Pointer:

org.eclipse.xtext.tutorial.survey.formatting.SurveyFormatter

Content
Assist

Quick
Fixes

Syntax
Highlighting

Outline

Template
Proposals

Formatting

etc.

Rename Refactoring

Match Braces

Mark Occurrences

Find References

Indexing

Error Markers

Toggle Comment

Open Model Element Dialog

Open Declaration

Code Folding

Autobuild

Open Source/Generated File

Hovers

Autoedit

Available Technology

- **Xtype:** Let your Language reference Java elements
- **Xbase:** Reuse an Expression Language
 - You've just used it as part of Xtend
 - Try it by playing XRobots at the itemis booth!
 - check out "Seven JVM Languages Built With Xbase"
- **Xcore:** A DSL to define Ecore models
- **Xpect:** A DSL to turn your DSL-files into Unit Tests

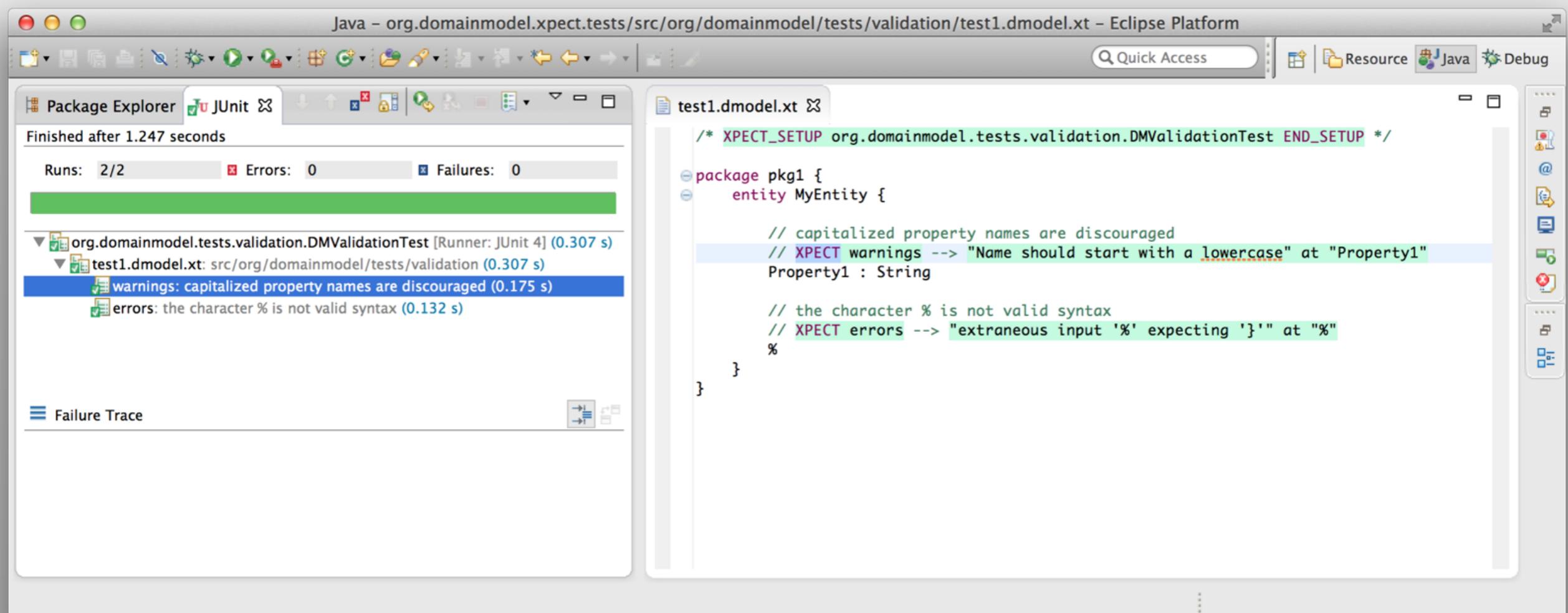
XtextRunner and test utils

→ ships with Xtext

Xpect: reusable Tests as Library for...

→ available on GitHub

- Parser and AST
 - Code generators
 - Validation
 - Linking
- Scoping
 - ResourceDescriptions
 - JvmModellInferer
 - AST Modifications via textual diff



Outlook: Platforms

- Xtext in the Web
- based on Eclipse Orion
- Try it by playing XRobot at the itemis booth!
- Xtext in IntelliJ
- Demo!

Further Steps

Install via Update Site or Download Eclipse DSL Distribution

What others have built with Xtext:

<http://www.eclipse.org/Xtext/community.html>

Web Forum:

<https://www.eclipse.org/forums/index.php/f/27/>

Professional Support:

itemis

Book: Implementing Domain-Specific Languages with Xtext and Xtend by Lorenzo Bettini



XtextCON



2014

XtextCON 2014 on Vimeo
Hello again! Want more ways to share and show your love on Vimeo? Join free
vimeo Join Log In Create Watch Upload Search
XtextCON
XtextCON - Scoping, Linking and Indexing - Moritz Eysholdt from Xtext Team 4 months ago
Scoping, Linking and Indexing
Moritz Eysholdt, itemis AG
58:51 HD
Follow
XtextCON 2014
Created by Xtext Team PLUS 4 months ago
XtextCON is the place for new users to learn Xtext quickly and for existing users to understand advanced use cases. Get in touch with Xtext users and the developers behind it. Discuss bugzillas and feature requests and get your individual problems solved.
See
+ More
XtextCON 2014
Display a menu

2015

Mark the Date!
May 18 - 20



XRobots

Program your robot in a web browser. Teach it sumo wrestling.
Challenge others and watch the show at the itemis booth.

join the game at
xrobots.itemis.de

Xtext, here @ EclipseCon

Play XRobot!		itemis booth
Diagrams, Xtext and UX	Jan Koehlein	Theater Stage Tuesday, 16:45
Enhanced Tooling for Modern Eclipse Plugin Development	Mikael Barbero	Silchersaal Thursday, 13:30
Maintain the Abstraction: Debuggers for your interpreted DSLs	Stephan Herrmann	Seminarraum 5 Tuesday, 15:00
New & Noteworthy in Xtend	Sebastian Zarnekow Stefan Oehme	Theater Tuesday, 16:45
New & Noteworthy in Xtext	Holger Schill Jan Koehlein	Theater Wednesday, 16:15
Sirius + Xtext = ❤	Cedric Brun	Theater Wednesday, 13:45
Xbase - The Complete Guide	Sebastian Zarnekow	Theater Stage Thursday, 13:30
Xtext in a Polyglot World	Sebastian Zarnekow	Theater Wednesday, 17:00