When Modeling meets Productivity

Sven Efftinge - itemis
I ❤️ Eclipse JDT
I ❤️ GIT
So what's the Problem?
It’s the Language not the Tooling!
Level of Abstraction
Reuse existing, proven technology and apply the lessons learned
Eclipse IDE Framework (JFace Text, etc.)
Java Development Tools (JDT)

Eclipse IDE Framework (JFace Text, etc.)
Java Development Tools (JDT)

Eclipse IDE Framework (JFace Text, etc.)

Eclipse Modeling Framework (EMF)
Xtext is Highly Adaptable
Advanced Language-Specific Editing Features by Default
Advanced Language-Specific Editing Features by Default

Syntax Coloring
Advanced Language-Specific Editing Features by Default

Syntax Coloring

Wizards
Advanced Language-Specific Editing Features by Default

Content Assist

Syntax Coloring

Wizards
Advanced Language-Specific Editing Features by Default

Content Assist

Syntax Coloring

Wizards

Code Folding
Advanced Language-Specific Editing Features by Default

Content Assist

Find References

Wizards

Syntax Coloring

Code Folding
Advanced Language-Specific Editing Features by Default

- Content Assist
- Find References
- Syntax Coloring
- Global Indexing
- Code Folding
- Wizards
Advanced Language-Specific Editing Features by Default

- Go To Declaration
- Global Indexing
- Content Assist
- Syntax Coloring
- Code Folding
- Find References
- Wizards
Advanced Language-Specific Editing Features by Default

Go To Declaration

Global Indexing

Content Assist

Find References

Syntax Coloring

Wizards

Bracket Matching

Code Folding

Tuesday, November 2, 2010
Advanced Language-Specific Editing Features by Default

- Go To Declaration
- Global Indexing
- Content Assist
- Syntax Coloring
- Find References
- Auto Editing
- Code Folding
- Bracket Matching
- Wizards

Tuesday, November 2, 2010
Advanced Language-Specific Editing Features by Default

Go To Declaration

Global Indexing

Content Assist

Find References

Syntax Coloring

Incremental Builders

Auto Editing

Bracket Matching

Code Folding

Wizards
Advanced Language-Specific Editing Features by Default

Content Assist

Find References

Incremental Builders

Wizards

Syntax Coloring

Global Indexing

Outline View

Go To Declaration

Auto Editing

Bracket Matching

Code Folding
Advanced Language-Specific Editing Features by Default

Syntax Coloring
Auto Editing
Code Folding
Outline View
Go To Declaration
Global Indexing
Content Assist
Find References
Validation
Incremental Builders
Bracket Matching
Auto Editing
Syntax Coloring
Wizards
Advanced Language-Specific Editing Features by Default

- Template Proposals
- Go To Declaration
- Content Assist
- Global Indexing
- Outline View
- Find References
- Syntax Coloring
- Validation
- Incremental Builders
- Auto Editing
- Wizards
- Bracket Matching
- Code Folding
Advanced Language-Specific Editing Features by Default

- Template Proposals
- Go To Declaration
- Content Assist
- Global Indexing
- Quick Fixes
- Outline View
- Find References
- Syntax Coloring
- Validation
- Incremental Builders
- Auto Editing
- Wizards
- Bracket Matching
- Code Folding

Tuesday, November 2, 2010
But hey, that’s all text!
How about another syntax?
/* Copyright (c) 2008, 2010 items AG and others.
 * All rights reserved. This program and the accompanying materials
 * are made available under the terms of the Eclipse Public License
 * which accompanies this distribution, and is available at
 */

grammar org.eclipse.xtext.Xtext with org.eclipse.xtext.common.Terminals

import "classpath:/org/eclipse.xtext/Xtext.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

Grammar:
   'grammar' name=GrammarID (with 'usedGrammars=[Grammar|Grammar])
   (definesHiddenTokens='hidden') {' (hiddenTokens=[AbstractRule]
   metamodelDeclarations=AbstractMetamodelDeclaration* (rules=AbstractRule)*
   }

GrammarID returns ecore::EString:
   ID ('.' ID)*

AbstractRule : ParserRule | TerminalRule | EnumRule;

AbstractMetamodelDeclaration :
   GeneratedMetamodel | ReferencedMetamodel;
   /*
   * constraint: typeSelect(GeneratedMetamodel).size() == typeSelect(ReferencedMetamodel).size()
   * generated metamodels have to have different aliases
   */

GeneratedMetamodel :
   'generate' name=ID ePackage=[ecore::EPackage|STRING] (as alias=ID)?
   /*
   * referenced metamodels may share aliases with other referenced metamodels
   */

ReferencedMetamodel :
   'import' ePackage=[ecore::EPackage|STRING] (as alias=ID)?

ParserRule :
   name=ID (returns type=TypeRef)? (definesHiddenTokens='hidden')
   alternatives=Alternatives
entity Person {
    property givenName : String
    property surName : String
    addresses refs many Address
}

entity Address {
    property street : String
    property number : String
    property city : String
}

entity Order {
    orderItems refs many OrderItem
    property orderDate : String
    property deliveryDate : String
}

entity OrderItem {
    property name : String
    property id : Number
}

entity Customer extends Person {
    property customerID : String
    orders refs many Order
}
Xtext is a scalable, easy to use language development framework.
Xtext is a scalable, easy to use language development framework based on Eclipse and EMF.
Xtext is a scalable, easy to use language development framework based on Eclipse and EMF bringing JDT-like editing experience to your language.