JavaCro'16

How to develop your own DSLs in Eclipse using Xtend and Xtext ?

Daniel Strmečki | Software Developer

19.05.2016.

BUSINESS WEB APPLICATIONS

EVOLVA

Content

1. DSLs
   - What are DSLs?
   - Kinds of DSLs?
   - Their usage and benefits…
2. Xtend
   - What kind of language is Xtend?
   - What are its benefits compared to classic Java?
3. Xtext
   - What is a language development framework?
   - How to develop you own DSL?
4. Short demo
5. Conclusion

DSL acronym

Not in a Telecommunications context: Digital Subscriber Line

But in a Software Engineering context: Domain Specific Languages

Domain Specific Languages

General-Purpose Languages (GPLs)

- Languages broadly applicable across application domains
- Lack specialized features for a particular domain
- XML, UML, **Java**, C++, Python, PHP…

Domain specific languages (DSLs)

- Languages tailored to for an application in a specific domain
- Provide substantial gains in expressiveness and ease of use in their domain of application
- SQL, HTML, CSS, Logo, Mathematica, Marcos, Regular expressions, Unix shell scripts, MediaWiki, LaTeX…

Domain Specific Languages

A novelty in software development

- DSL for programming numerically controlled machine tools, was developed in 1957–1958
- BNF dates back to 1959 (M. Mernik)

The line between DSL and GPL

- A language may have specialized features for a particular domain but be applicable more broadly
- A language may be capable of broad application but in practice used primarily for a specific domain
- In combination with an **application library**, any GPL can act as a DSL. Most DSLs never get beyond the application library stage (M. Mernik)

Domain Specific Languages

The kind of DSL

- Domain Specific **Markup** Languages (XML, HTML)
- Domain Specific **Modeling** Languages (UML, BPMN)
- Domain Specific **Programming** Languages (…)

Design and implementation

- **External DSL**
  - A free-standing DSL
  - Designed to be independent of any GPL
- **Internal or embedded DSL**
  - Implemented using a host language (e.g. Java)
  - The goal is to exploit the metaprogramming capabilities of the host language

Domain Specific Languages

Some benefits

- A higher abstraction level
  - The same functionality achieved with **less code**
- Focusing on the problem and not on the solution
- Easier **modifications**
- Simplified **maintenance**
- **Validation** at the domain level
  - The code is understood and validated by domain experts
- Shift the development to **domain experts**
  - Business information systems development can be partially shifted from traditional software developers to the typically larger group of domain-experts

Domain Specific Languages

Develop your own DSLs

- DSL developments requires both domain and language development **knowledge** and **skills**
  - Both technical experts and non-technical domain experts need to be involved in DSL development
- DSL development is **hard** and **expensive**
  - Designing, implementing, and maintaining a DSL as well as the tools required to develop with it is costly
    - Developing an IDE for your DSL
    - Developing the **tool support** required to ensure quality and reliability of software developed by end-user programmers (M. Mernik)

XTEND

Xtend is a statically-typed **programming language** which meets the requirements for code generation and translates to comprehensible Java source code

Eclipse project

- Open source under the Eclipse Public License
- Can be compiled and run independent of Eclipse

XTEND

Type of language

- Primarily an **object-oriented** language, but it also integrates features from functional programming
- **Statically-typed** programming language

A JVM language

- Syntactically and semantically roots in **Java** but focuses on a more concise syntax and **additional functionality**
- It uses **Java's type system** without modifications
- It translates to comprehensible **Java source code** and thereby seamlessly integrates with all existing **Java libraries**
- It has no interoperability issues with **Java**

XTEND

Philosophy

- Java, as one of the most popular programming languages, has grown an enormous ecosystem of libraries and tools
- Java's syntax is quite verbose, and **some concepts are missing and added very slowly**
- Xtend gets the best of Java, but **kills the syntactic noise** and adds essential **new features** for better readability and more high level code
- Java code and Xtend code can be **mixed** in a same project
- Xtend can use **Java libraries**, call **Java functions**, use **Java objects**…
- Xtend can be extended by means of libraries

XTEND

A Hello World

› Eclipse will automatically translate Xtend to Java code when you save your .xtend files

```
class HelloWorld {
  def static void main(String[] args) {
    println("Hello World")
  }
}
```

Type safety

```
def static void main(String[] args) {
    val i = 5;
    var string = method(i);
    print(string)
}
def static method(int i) {
    return "This is a sting " + i;
}
```

XTEND

Extensions

› Enhanced closed types with new functionality

```
def static void main(String[] args) {
    val hello = "hello".toUpperCase;
    hello.doSomething
    var list = newArrayList("a", "b", "c")
    list.forEach[ element, index |
        println(hello + ":" + element)
    ]
}
def static doSomething(Object o) {}
```

Lambda Expressions

› Lambdas for Java 8+ and anonymous classes for Java 7-

```
val toUpperCaseFunction = [ String s | s.toUpperCase ]
var someStrings = new ArrayList<String>();
Collections.sort(someStrings, [ a, b |
  a.length - b.length
])
```

XTEND

Type casts

› Like casts in Java, but with more readable syntax

```
var obj = new Object();
var i = obj as MyInterface
var j = 42 as Integer
```

Equality operators

› Equals operators (==,!=) are bound to Object.equals

```
if (name == 'Homer')
    println('Hello Homer')
```

Null-safe calls and Groovy Elvis operator

```
hello?.doSomething // null check
val salutation = name ?: 'Sir/Madam'
```

XTEND

Arithmetic operators

› Operators are not limited to operations on certain types

```
val x = 2.71BD
val y = 3.14BD
// calls BigDecimalExtension.operator_plus(x,y)
val sum = x + y
```

With operator

› Initialize objects in subsequent lines of code

```
val person = new Person => [
      firstName = 'Homer'
      lastName = 'Simpson'
      address = new Address => [
              street = '742 Evergreen Terrace'
              city = 'SpringField'
      ]
]
```

XTEND

Pair operator

› Use a pair of two elements locally without introducing a new structure

```
val nameAndAge = 'Homer' -> 42
```

Range operators

```
// iterate the list forwards
for (i : 0 ..< someStrings.size) {
    val element = someStrings.get(i)
    print(element)
}
// or backwards
for (i : someStrings.size >.. 0) {
    val element = someStrings.get(i)
    print(element)
}
```

XTEND

Template expressions

› A template expression can span multiple lines and allows readable string concatenation with loops and conditions

```
def someHTML(List<String> names)
    '''
      <html>
        <body>
          <p>
            «FOR name : names»
            Welcome «name»
              «IF name == "Daniel"»
              Role: <b>Administrator</b>
              «ELSE»
              Role <b>User</b>
              «ENDIF»
            «ENDFOR»
          </p>
        </body>
      </html>
    '''
```

XTEXT

Xtext is a programming **language development framework** that provides a powerful grammar language used to generate a full language infrastructure

Eclipse project

- Open source under the Eclipse Public License
- Based on **Eclipse Modeling Framework** (EMF), **Xtend** and **Google Guice**

XTEXT

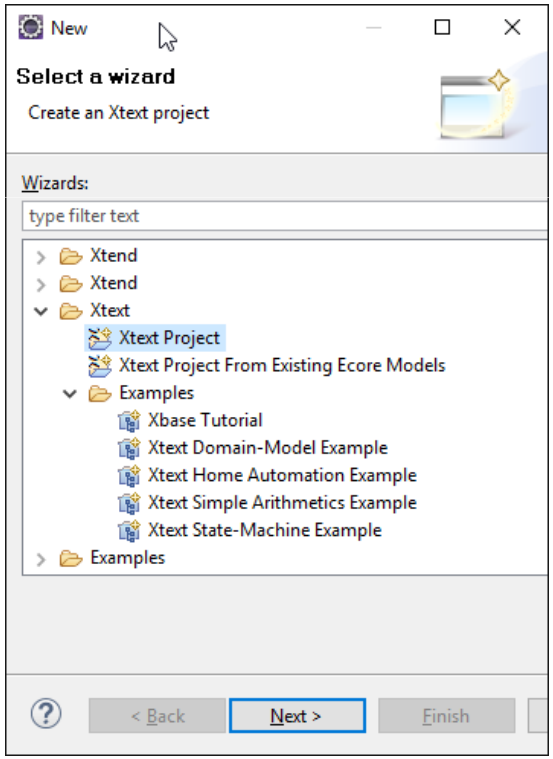
Language Development Framework

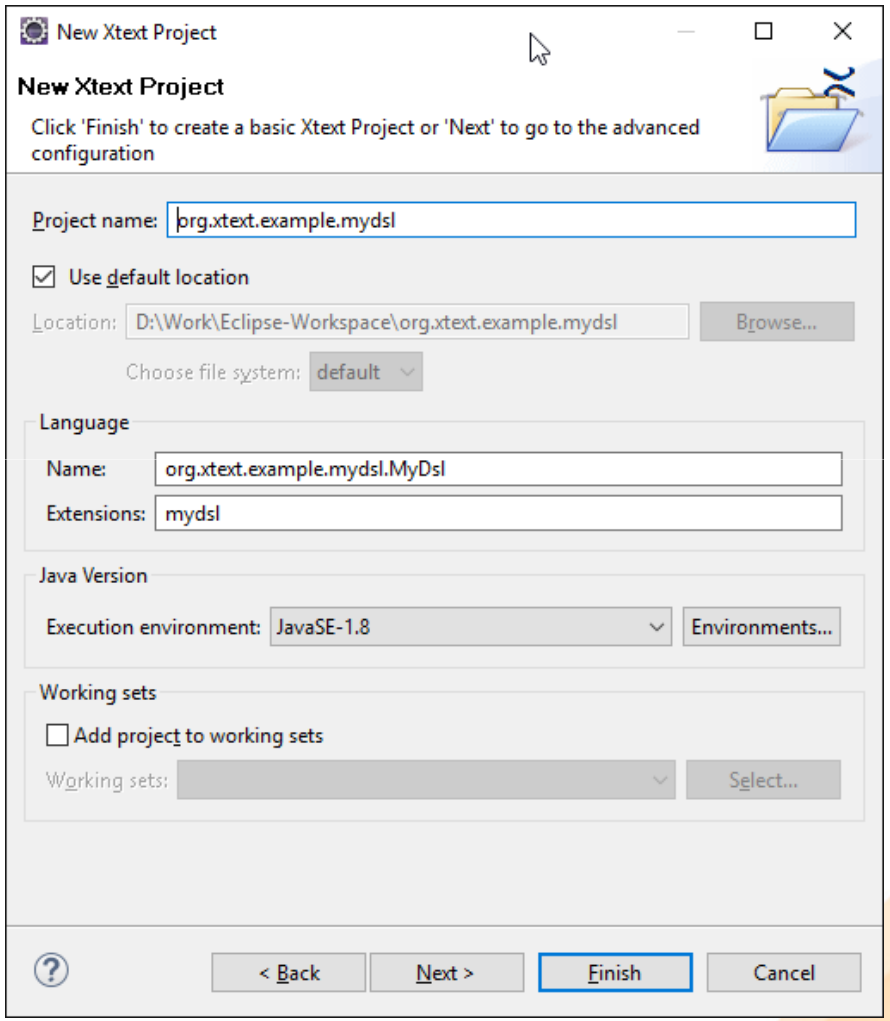
- Xtext allows language developers to create a sophisticated Eclipse-based development environment providing editing experience known from modern Java IDEs in a surprisingly **short amount of time**
- It provides a set of DSLs and modern APIs to describe the different aspects of **your programming language**
- Based on the language description it gives a full implementation of that language running on the JVM
  - parser and a type-safe abstract syntax tree (AST)
  - serializer and code formatter
  - scoping framework and linking
  - compiler checks and static analysis
  - code generator or interpreter

XTEXT

Your own DSL

- Eclipse
  - Xtext project





XTEXT

Extended Backus–Naur Form (EBNF) grammar

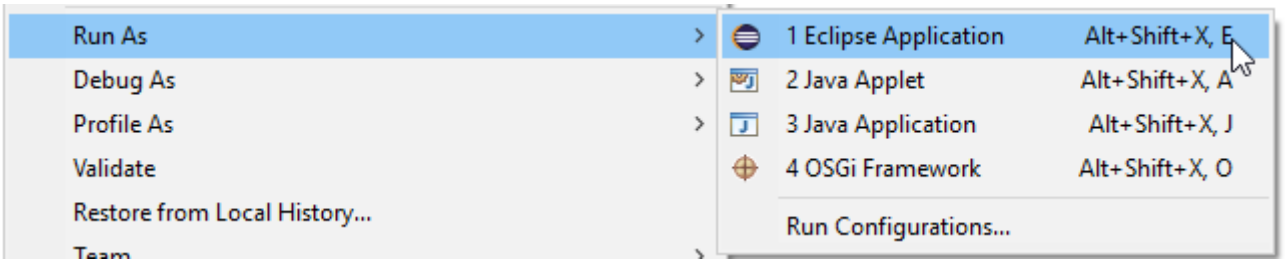
```
grammar org.eclipse.xtext.example.homeautomation.RuleEngine
with org.eclipse.xtext.xbase.Xbase

import "http://www.eclipse.org/xtext/xbase/Xbase" as xbase
generate ruleEngine "http://www.eclipse.org/Xtext/example/RuleEngine"

Model:
    declarations+=Declaration*;

Declaration:
    Device | Rule;

Device:
    'Device' name=ID 'can' 'be'
        (states+=State (',' states+=State)*)?;

State:
    name=ID ;

Rule:
    'Rule' description=STRING
        'when' deviceState=[State|QualifiedName]
        'then' thenPart=XBlockExpression;
```

XTEXT

Generate and test your DSL

› Generate the DSL implementation for your grammar

› Test your DSL in a new instance of Eclipse IDE



XTEXT

Advanced features

› Formatting

```
def dispatch void format(Rule rule, extension IFormattableDocument document) {
    rule.regionFor.feature(RULE__DESCRIPTION).surround[oneSpace]
    rule.regionFor.feature(RULE__DEVICE_STATE).surround[oneSpace]
    rule.thenPart.format.prepend[newLine]
}
```

› Validation

```
@Check
def checkStatesNotEmpty(Device device) {
    if (device.states.empty) {
        error('''The device "«device.name»" must have at least one state.''',
            device, DEVICE__NAME
        )
    }
}
```

XTEXT

Code generation

```
class DBDslGenerator extends AbstractGenerator {

    override doGenerate(Resource input, IFileSystemAccess2 fsa,
        IGeneratorContext context) {
        fsa.generateFile('greetings.txt', 'My table names: ' +
            input.allContents
                .filter(typeof(Table))
                .map[name]
                .join(', '))
    }

}
```

More features

› ...

JavaCro'16

A SHORT DEMO

EVOLVA

Conclusion

- DSLs can be developed when we are in the need for higher abstraction levels and expressiveness in a certain domain
  - The same functionality achieved with **less code**
- Xtend is a statically-typed programming language suited for writing **code generators** that translates to comprehensible **Java source code**
- Xtext is a language development framework that provides a **powerful grammar language** used to generate a **full language infrastructure** for your custom DSLs
- In Software engineering there are no universal solutions, as it entails creative processes which are always critically dependent on the **unique abilities** of the **creative people** who perform them (M. A. Musen)

JavaCro'16

Thanks for your attention

www.evolva.hr

daniel.strmecki@evolva.hr