
Domain Specific Languages

FrOSCon 2012

Tobias Schlitt (@tobySen)

Gordon Oheim (@go_oh)

2012-08-25

Outline

About us

DSL WTF?

External DSL

Internal DSL

Abusing other languages

Summary

About us

- ▶ Tobias Schlitt (**Toby**)



About us

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP



About us

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts



About us

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Software Architecture, Design, Testing ...



About us

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Software Architecture, Design, Testing ...

Co-founder of



About us

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Software Architecture, Design, Testing ...

Co-founder of



**Helping teams to create
high quality web
applications.**

About us

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Software Architecture, Design, Testing ...

Co-founder of



**Helping teams to create
high quality web
applications.**

<http://qafoo.com>

About us

- ▶ **Gordon** Oheim



About us

- ▶ **Gordon** Oheim
- ▶ Professional PHP developer
- ▶ More than 10 years experience



About us

- ▶ **Gordon** Oheim
- ▶ Professional PHP developer
- ▶ More than 10 years experience
- ▶ OOA, OOD, OOP lover
- ▶ Interested in agile and lean



About us

- ▶ **Gordon** Oheim
- ▶ Professional PHP developer
- ▶ More than 10 years experience
- ▶ OOA, OOD, OOP lover
- ▶ Interested in agile and lean
- ▶ PHP Documentation Team
- ▶ Stack overflow contributor



About us

- ▶ **Gordon** Oheim
- ▶ Professional PHP developer
- ▶ More than 10 years experience
- ▶ OOA, OOD, OOP lover
- ▶ Interested in agile and lean
- ▶ PHP Documentation Team
- ▶ Stack overflow contributor



Freelance Developer

Outline

About us

DSL WTF?

External DSL

Internal DSL

Abusing other languages

Summary

A common DSL

```
1 (
2     http://           # match hyperlinks
3     (?:www\.)?       # optional www
4     stackoverflow.com/ # pointing to stackoverflow
5     q(?:uestions)?/  # with path q or questions
6     (?P<qid>\d+)      # and get the Question ID
7 ) xiu
```


PCRE

- ▶ Only for text matching
- ▶ Limited expressability
- ▶ Embedded into other languages (e.g. PHP)

PCRE

- ▶ Only for text matching
- ▶ Limited expressability
- ▶ Embedded into other languages (e.g. PHP)

... concise and flexible means to “match” (specify and recognize) strings of text, such as particular characters, words, or patterns of characters.

–Wikipedia

Another well-known DSL

```
1 SELECT COUNT(course.person_id) AS count_persons ,  
   professor.name  
2 FROM professor  
3 LEFT OUTER JOIN course  
4   ON professor.id = course.prof_id  
5 GROUP BY professor.id
```

SQL

- ▶ Only for RDBMS querying / manipulation
- ▶ Limited expressability
- ▶ Embedded into other languages (e.g. PHP)

SQL

- ▶ Only for RDBMS querying / manipulation
- ▶ Limited expressability
- ▶ Embedded into other languages (e.g. PHP)

... a special-purpose programming language designed for managing data in relational database management systems...

–Wikipedia

What is a DSL?

Domain-specific language (noun): a computer programming language of limited expressiveness focused on a particular domain.

–Martin Fowler

Do you know more?

Do you know more?

XPath

Do you know more?

ant

XPath

Do you know more?

CSS

ant

XPath

Do you know more?

graphviz

CSS

ant

XPath

Do you know more?

graphviz

CSS

Java

ant

XPath

Do you know more?

graphviz

CSS

Java

ant

XPath

Java is a Domain Specific Language to compile large XML files into stack traces

Why create a DSL?

Why create a DSL?

- ▶ Easy to read and use

Why create a DSL?

- ▶ Easy to read and use
- ▶ **Business communication**

Why create a DSL?

- ▶ Easy to read and use
- ▶ Business communication
- ▶ Simplify complex model adjustments

Why create a DSL?

- ▶ Easy to read and use
- ▶ Business communication
- ▶ Simplify complex model adjustments
- ▶ **Ease repetetive tasks**

Why create a DSL?

- ▶ Easy to read and use
- ▶ Business communication
- ▶ Simplify complex model adjustments
- ▶ Ease repetetive tasks
- ▶ **Expressive and noiseless**

Why create a DSL?

- ▶ Easy to read and use
- ▶ Business communication
- ▶ Simplify complex model adjustments
- ▶ Ease repetetive tasks
- ▶ Expressive and noiseless
- ▶ **Powerful configuration**

Outline

About us

DSL WTF?

External DSL

Internal DSL

Abusing other languages

Summary

Example: Gherkin

Gherkin is a Business Readable DSL created especially for behavior descriptions.

–Gherkin website

Example: Gherkin

Gherkin is a Business Readable DSL created especially for behavior descriptions.

–Gherkin website

- ▶ Behavior Driven Development (BDD)
- ▶ Write test cases your business can read
- ▶ Create your own DSL based on Gherkin

Example: Gherkin

1 Feature: Serve coffee

2 In order to earn money

3 Customers should be able to
4 buy coffee at all times

5
6 Scenario: Buy last coffee

7 Given there are 1 coffees left in the machine

8 And I have deposited 1 dollar

9 When I press the coffee button

10 Then I should be served a coffee

Example: Behat

- ▶ BDD framework written in PHP
- ▶ Inspired by Cucumber
- ▶ Runs tests written in Gherkin
- ▶ <http://behat.org/>

Example: Behat

```
1 class FeatureContext extends BehatContext
2 {
3     /**
4     * @Given /^I am in a directory "([^"]*)"$/
5     */
6     public function iAmInADirectory($dir)
7     {
8         if (!file_exists($dir)) {
9             mkdir($dir);
10        }
11        chdir($dir);
12    }
13 }
```

Building an external DSL

1. Domain Language
2. Grammar (BNF)
3. Lexer
4. Parser (or Parser-Generator)
5. ...

Gherkin Parser

```
50 public function parse($input, $file = null)
51 {
52     $this->file = $file;
53
54     try {
55         $this->lexer->setInput($input);
56     } catch (LexerException $e) {
57         throw new ParserException(
58             sprintf('Lexer exception "%s" thrown for file %s', $e->getMessage(), $file)
59         );
60     }
61
62     $this->lexer->setLanguage($language = 'en');
63     $languageSpecifierLine = null;
64
65     $feature = null;
66     while ('EOS' !== ($predicted = $this->predictTokenType())) {
67         if ('Newline' === $predicted || 'Comment' === $predicted) {
68             $this->lexer->getAdvancedToken();
69         } elseif ('Language' === $predicted) {
70             $token = $this->expectTokenType('Language');
71             $language = $token->value;
72
73             if (null === $languageSpecifierLine) {
74                 // Reparse input with new language
75                 $languageSpecifierLine = $token->line;
76                 $this->lexer->setInput($input);
77                 $this->lexer->setLanguage($language);
78             } elseif ($languageSpecifierLine !== $token->line) {
79                 // Language already specified
80                 throw new ParserException(sprintf(
81                     'Ambiguous language specifiers on lines: %d and %d%s',
82                     $languageSpecifierLine,
83                     $token->line,
```



Outline

About us

DSL WTF?

External DSL

Internal DSL

Abusing other languages

Summary

Example: Interface Distiller

Allows you to create Interfaces from existing classes

–Gordon

Example: Semantic Model 1/2

```
1 $reflector = new \ReflectionClass( 'ConcreteFoo' );
2
3 $methodIterator = new Filters \
4     NoImplementedMethodsIterator(
5     new Filters \ NoInheritedMethodsIterator (
6         new Filters \ NoOldStyleConstructorIterator (
7             new Filters \ NoMagicMethodsIterator (
8                 new \ArrayIterator ( $reflector ->
9                     getMethods () )
10             )
11         )
12     ),
13     $reflector
14 );
```

Example: Semantic Model 2/2

```
1 $distillate = new Distillate ;
2
3 $distillate ->setInterfaceName( 'FooInterface' );
4 $distillate ->setExtendingInterfaces( 'Iterator', '
    SeekableIterator' );
5 foreach ( $methodIterator as $method ) {
6     $distillate ->addMethod( $method );
7 }
8
9 $file = new \SplFileObject( 'FooInterface' );
10 $writer = new Distillate\Writer( $file );
11 $writer ->writeToFile( $distillate );
```

Pattern: Semantic Model

- ▶ "The Domain"

Pattern: Semantic Model

- ▶ "The Domain"
- ▶ Lots of Syntactic Noise
- ▶ No linguistic abstraction

Pattern: Semantic Model

- ▶ "The Domain"
- ▶ Lots of Syntactic Noise
- ▶ No linguistic abstraction
- ▶ **Basis for "the language"**

Pattern: Semantic Model

- ▶ "The Domain"
- ▶ Lots of Syntactic Noise
- ▶ No linguistic abstraction
- ▶ Basis for "the language"
- ▶ Follows general design principles

Example: Expression Builder

```
1 $distiller = new InterfaceDistiller;  
2 $distiller  
3     ->methodsWithModifiers(\ReflectionMethod::  
4         IS_PUBLIC)  
5     ->extendInterfaceFrom('Iterator', 'SeekableIterator'  
6         )  
7     ->excludeImplementedMethods()  
8     ->excludeInheritedMethods()  
9     ->excludeMagicMethods()  
10    ->excludeOldStyleConstructors()  
11    ->saveAs('FooInterface.php')  
12    ->distill('ConcreteFoo', 'FooInterface');
```

Pattern: Expression Builder

- ▶ "The language"

Pattern: Expression Builder

- ▶ "The language"
- ▶ Capture essential usage
- ▶ Hide complexity and mechanics

Pattern: Expression Builder

- ▶ "The language"
- ▶ Capture essential usage
- ▶ Hide complexity and mechanics
- ▶ **Effectively: Semantic Facade**

Pattern: Expression Builder

- ▶ "The language"
- ▶ Capture essential usage
- ▶ Hide complexity and mechanics
- ▶ Effectively: Semantic Facade
- ▶ **Fluent Interface**
- ▶ **Pragmatic implementation**

Example: Different Syntax

```
1  phpdistill --bootstrap ./bootstrap.php \  
2  --methodsWithModifiers 256 \  
3  --extendInterfaceFrom Iterator , \  
4  SeekableIterator \  
5  --excludeImplementedMethods \  
6  --excludeInheritedMethods \  
7  --excludeMagicMethods \  
8  --excludeOldStyleConstructors \  
9  --saveAs FooInterface.php ConcreteFoo  
   FooInterface
```

Additional example: Zeta SQL Abstraction

```
1 <?php
2
3 $selectQuery
4     ->update( 'ezcontentobject_attribute' )
5     ->set(
6         'language_id',
7         $selectQuery->bindValue( $newLanguageMask )
8     )
9     ->where(
10        $selectQuery->expr->IAnd(
11            $selectQuery->expr->eq(
12                'contentobject_id',
13                $selectQuery->bindValue( $contentId )
14            ),
15            $selectQuery->expr->eq(
16                'version',
17                $selectQuery->bindValue( $versionNo )
18            )
19        )
20    );
```

Additional example: Mockery

```
1 <?php
2
3 $mock = M::mock( 'Engineering' );
4 $mock->shouldReceive( 'disengageWarp' )
5     ->once()->ordered();
6 $mock->shouldReceive( 'divertPower' )
7     ->with(0.40, 'sensors')->once()->ordered();
8 $mock->shouldReceive( 'divertPower' )
9     ->with(0.30, 'auxengines')->once()->ordered();
10 // ...
11
12 $starship = new Starship($mock);
13 $starship->enterOrbit();
```

Outline

About us

DSL WTF?

External DSL

Internal DSL

Abusing other languages

Summary

Example: Code browser

Renders code into a browsable HTML presentation

–Toby

Example: Code browser DSL

```
1 base:    "./"
2 -
3   file:   "example3-behat.php"
4   display: "2,8,9-21"
5   highlights:
6       1:  "9,10,21"
7       2:  "11-13"
8       3:  "16-19"
9 -
10  file:   "example7-interface-distiller.php"
11  display: "3-7,11,12"
12  highlights:
13      1:  "3"
14      2:  "11,12"
```

Example: Code browser DSL

```
1 base:      "./"
2 -
3   file:     "example3-behat.php"
4   display:  "2,8,9-21"
5   highlights:
6       1:    "9,10,21"
7       2:    "11-13"
8       3:    "16-19"
9 -
10  file:     "example7-interface-distiller.php"
11  display:  "3-7,11,12"
12  highlights:
13      1:    "3"
14      2:    "11,12"
```

Example: Code browser DSL

```
1 base:      "./"
2 -
3   file:     "example3-behat.php"
4   display:  "2,8,9-21"
5   highlights:
6       1:    "9,10,21"
7       2:    "11-13"
8       3:    "16-19"
9 -
10  file:     "example7-interface-distiller.php"
11  display:  "3-7,11,12"
12  highlights:
13      1:    "3"
14      2:    "11,12"
```

Example: Code browser DSL

```
1 base:    "./"
2 -
3   file:   "example3-behat.php"
4   display: "2,8,9-21"
5   highlights:
6       1:  "9,10,21"
7       2:  "11-13"
8       3:  "16-19"
9 -
10  file:    "example7-interface-distiller.php"
11  display: "3-7,11,12"
12  highlights:
13      1:  "3"
14      2:  "11,12"
```

Example: Code browser DSL

```
1 base:    "./"
```

```
2 -
```

```
3   file:   "example3-behat.php"
```

```
4   display: "2,8,9-21"
```

```
5   highlights:
```

```
6     1:    "9,10,21"
```

```
7     2:    "11-13"
```

```
8     3:    "16-19"
```

```
9 -
```

```
10  file:   "example7-interface-distiller.php"
```

```
11  display: "3-7,11,12"
```

```
12  highlights:
```

```
13    1:    "3"
```

```
14    2:    "11,12"
```

Example: Code browser Semantic Model

```
1  $codeManager = new CodeManager( ..DIR.. );
2
3  $slideGenerator = new ListingSlideGenerator(
4      $codeManager,
5      'example3-behat.php',
6      new DisplayFilter(
7          new Tools\Range(
8              2, 8, range( 9, 21 )
9          )
10     ),
11     new HighlightDefinition(
12         array(
13             array(
14                 new Tools\Range( 1 ),
15                 new Tools\Range( 9, 10, 21 )
16             ),
17             array(
18                 new Tools\Range( 2 ),
19                 new Tools\Range( range( 11, 13 ) )
20             ),
21             array(
22                 new Tools\Range( 3 ),
23                 new Tools\Range( range( 16, 19 ) )
24             ),
25         )
26     )
27 );
```

Example: Code browser Semantic Model

```
1 $codeManager = new CodeManager( __DIR__ );
2
3 $slideGenerator = new ListingSlideGenerator(
4     $codeManager,
5     'example3-behat.php',
6     new DisplayFilter(
7         new Tools\Range(
8             2, 8, range( 9, 21 )
9         )
10    ),
11    new HighlightDefinition(
12        array(
13            array(
14                new Tools\Range( 1 ),
15                new Tools\Range( 9, 10, 21 )
16            ),
17            array(
18                new Tools\Range( 2 ),
19                new Tools\Range( range( 11, 13 ) )
20            ),
21            array(
22                new Tools\Range( 3 ),
23                new Tools\Range( range( 16, 19 ) )
24            ),
25        )
26    )
27 );
```

Example: Code browser Semantic Model

```
1  $codeManager = new CodeManager( __DIR__ );
2
3  $slideGenerator = new ListingSlideGenerator(
4      $codeManager,
5      'example3-behat.php',
6      new DisplayFilter(
7          new Tools\Range(
8              2, 8, range( 9, 21 )
9          )
10     ),
11     new HighlightDefinition(
12         array(
13             array(
14                 new Tools\Range( 1 ),
15                 new Tools\Range( 9, 10, 21 )
16             ),
17             array(
18                 new Tools\Range( 2 ),
19                 new Tools\Range( range( 11, 13 ) )
20             ),
21             array(
22                 new Tools\Range( 3 ),
23                 new Tools\Range( range( 16, 19 ) )
24             ),
25         )
26     )
27 );
```


Example: Code browser Semantic Model

```
1 $codeManager = new CodeManager( __DIR__ );
2
3 $slideGenerator = new ListingSlideGenerator(
4     $codeManager,
5     'example3-behat.php',
6     new DisplayFilter(
7         new Tools\Range(
8             2, 8, range( 9, 21 )
9         )
10    ),
11    new HighlightDefinition(
12        array(
13            array(
14                new Tools\Range( 1 ),
15                new Tools\Range( 9, 10, 21 )
16            ),
17            array(
18                new Tools\Range( 2 ),
19                new Tools\Range( range( 11, 13 ) )
20            ),
21            array(
22                new Tools\Range( 3 ),
23                new Tools\Range( range( 16, 19 ) )
24            ),
25        )
26    )
27 );
```

Example: Code browser Semantic Model

```
1 $codeManager = new CodeManager( __DIR__ );
2
3 $slideGenerator = new ListingSlideGenerator(
4     $codeManager,
5     'example3-behat.php',
6     new DisplayFilter(
7         new Tools\Range(
8             2, 8, range( 9, 21 )
9         )
10    ),
11    new HighlightDefinition(
12        array(
13            array(
14                new Tools\Range( 1 ),
15                new Tools\Range( 9, 10, 21 )
16            ),
17            array(
18                new Tools\Range( 2 ),
19                new Tools\Range( range( 11, 13 ) )
20            ),
21            array(
22                new Tools\Range( 3 ),
23                new Tools\Range( range( 16, 19 ) )
24            ),
25        )
26    )
27 );
```

Example: Code browser Semantic Model

```
1  $codeManager = new CodeManager( __DIR__ );
2
3  $slideGenerator = new ListingSlideGenerator(
4      $codeManager,
5      'example3-behat.php',
6      new DisplayFilter(
7          new Tools\Range(
8              2, 8, range( 9, 21 )
9          )
10     ),
11     new HighlightDefinition(
12         array(
13             array(
14                 new Tools\Range( 1 ),
15                 new Tools\Range( 9, 10, 21 )
16             ),
17             array(
18                 new Tools\Range( 2 ),
19                 new Tools\Range( range( 11, 13 ) )
20             ),
21             array(
22                 new Tools\Range( 3 ),
23                 new Tools\Range( range( 16, 19 ) )
24             ),
25         )
26     )
27 );
```

Outline

About us

DSL WTF?

External DSL

Internal DSL

Abusing other languages

Summary

DSL vs. Configuration

Is every configuration a DSL?

DSL vs. Configuration

Is every configuration a DSL?

- ▶ Simple: No.

Is every configuration a DSL?

- ▶ Simple: No.
- ▶ Missing semantics
- ▶ Missing expressiveness

Is every configuration a DSL?

- ▶ Simple: No.
- ▶ Missing semantics
- ▶ Missing expressiveness
- ▶ But it's a fluid transition . . .

Summary

- ▶ DSLs can greatly enhance your live
 - ▶ Business communication
 - ▶ Simplify complex (repetitive?) tasks
 - ▶ Expressive and noiseless
 - ▶ Powerful configuration

Summary

- ▶ DSLs can greatly enhance your live
 - ▶ Business communication
 - ▶ Simplify complex (repetitive?) tasks
 - ▶ Expressive and noiseless
 - ▶ Powerful configuration
- ▶ No need for external DSL in 1st place

Summary

- ▶ DSLs can greatly enhance your live
 - ▶ Business communication
 - ▶ Simplify complex (repetitive?) tasks
 - ▶ Expressive and noiseless
 - ▶ Powerful configuration
- ▶ No need for external DSL in 1st place
 1. Create Semantic Model

Summary

- ▶ DSLs can greatly enhance your live
 - ▶ Business communication
 - ▶ Simplify complex (repetitive?) tasks
 - ▶ Expressive and noiseless
 - ▶ Powerful configuration
- ▶ No need for external DSL in 1st place
 1. Create Semantic Model
 2. Create an internal DSL
 3. Maybe abuse other language

Summary

- ▶ DSLs can greatly enhance your live
 - ▶ Business communication
 - ▶ Simplify complex (repetitive?) tasks
 - ▶ Expressive and noiseless
 - ▶ Powerful configuration
- ▶ No need for external DSL in 1st place
 1. Create Semantic Model
 2. Create an internal DSL
 3. Maybe abuse other language
 4. Create a (E)BNF
 5. Use a parser generator

Summary

- ▶ DSLs can greatly enhance your live
 - ▶ Business communication
 - ▶ Simplify complex (repetitive?) tasks
 - ▶ Expressive and noiseless
 - ▶ Powerful configuration
- ▶ No need for external DSL in 1st place
 1. Create Semantic Model
 2. Create an internal DSL
 3. Maybe abuse other language
 4. Create a (E)BNF
 5. Use a parser generator
- ▶ **DSLs are no golden hammer!**

Attribution

- ▶ Hammer photo CC-BY stebulus,
<https://secure.flickr.com/photos/stebulus/231058538/>