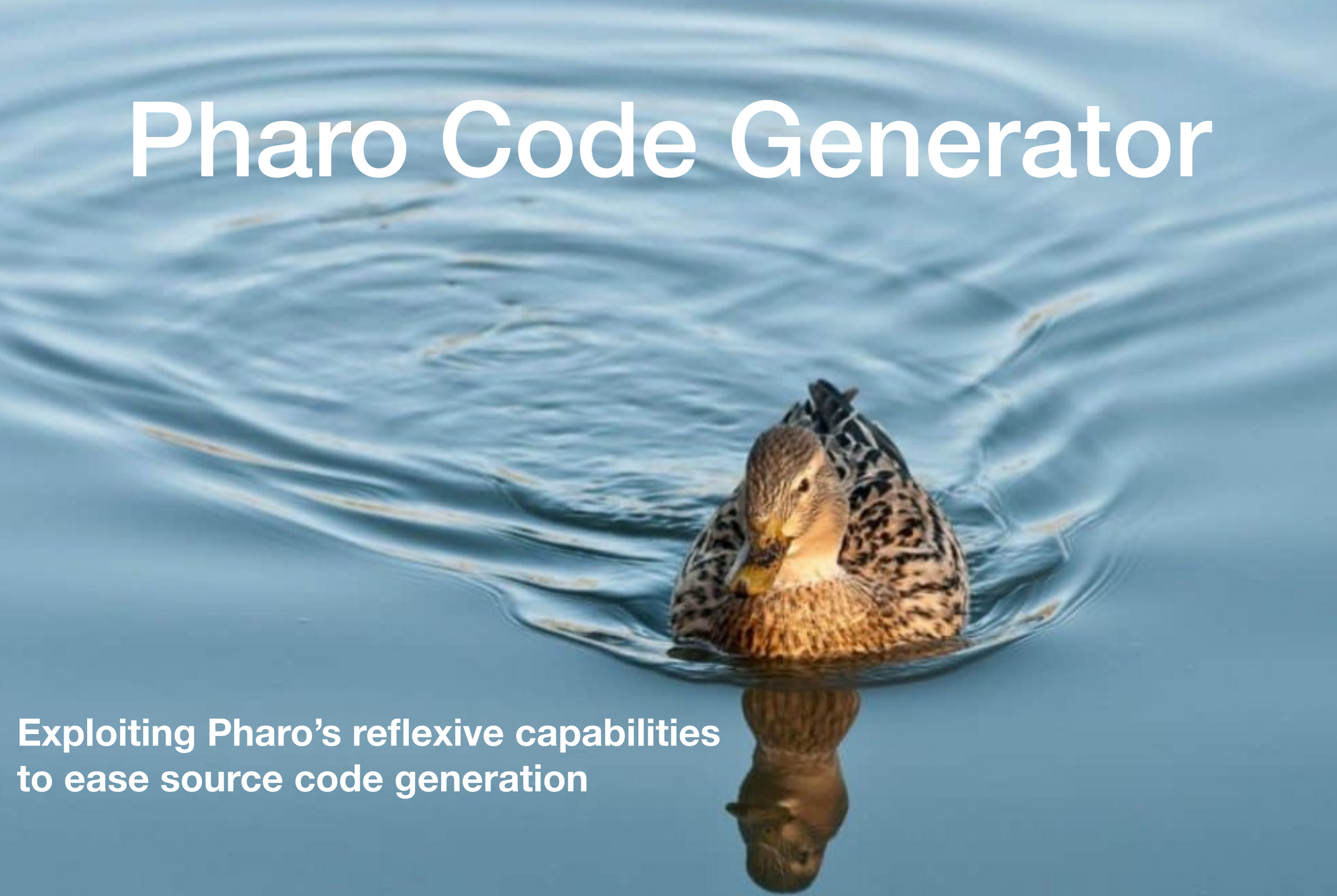
Pharo Code Generator



Exploiting Pharo's reflexive capabilities to ease source code generation

Julien Delplanque

julien.delplanque@inria.fr

Pharo Code Generator

- https://github.com/juliendelplanque/PharoCodeGenerator

- DSL to generate Pharo code in Pharo.

- In alpha stage :-)

- '**PCG**' will be used as shortcut for **P**haro **C**ode **G**enerator during this presentation.

How it works?

1. Describe the code to generate via the DSL

```
(PCGMethodNode selector: #answerToEverything)
    bodyBlock: [ :body |
        body << 42 asPCG returnIt ]
```

2. Execution generates PCG AST objects

PCG AST

3. Trigger translation to Pharo's AST

PCG visitors

- Checks validity
- Perform translation

Generates

Pharo's AST

4. Install generated code in the system

API overview

```
ast := (PCGMethodNode selector: #answerToEverything)
       bodyBlock: [ :body |
              body << 42 asPCG returnIt ]
```

generates

```
answerToEverything
  <generated>
  ^ 42
```

- `ast realAst` • Builds the Pharo AST corresponding to description made in DSL
- `ast checkAst` • Checks if the AST is ready for translation to Pharo AST. Raises an error if not.
- `ast sourceCode` • Returns a String holding source code resulting from generation.
- `ast withGeneratedPragma: true|false` • Adds or not <generated> pragma in method source code.

Template support

```
template := (PCGMethodNode selector: #answerSelector asPCGTemplateParameter)
        bodyBlock: [ :body |
                body << #answer asPCGTemplateParameter returnIt ]
```

```
(template substituteParametersWith: {
        #answerSelector -> #answerToEverything.
        #answer -> 42 asPCG }) sourceCode.
```

generates

```
answerToEverything
  <generated>
  ^ 42
```

```
(template substituteParametersWith: {
        #answerSelector -> #answerToEverythingInParallelUniverse.
        #answer -> 43 asPCG }) sourceCode
```

generates

```
answerToEverythingInParallelUniverse
  <generated>
  ^ 43
```

Do not start from scratch

If you saw a method in your project that can be used as a basis for code generation, PCG can help you

Do not start from scratch

1. Describe the code to generate via the DSL

```
(PCGMethodNode selector: #answerToEverything)
    bodyBlock: [ :body |
        body << 42 asPCG returnIt ]
```

2. Execution generates PCG AST objects

PCG AST

3. Trigger translation to Pharo's AST

PCG visitors

- Checks validity
- Perform translation

Generates

Pharo's AST

Generate PCG AST from Pharo AST

4. Install generated code in the system

Do not start from scratch

```
pcgAst := (Object >> #yourself) asPCGAST
```

pcgAst sourceCode —*generates*→

```
yourself
  <generated>
  ^ self
```

Do not start from scratch

```
pcgAst := (Object >> #yourself) asPCGAST
```

```
pcgAst meta sourceCode
```

generates

```
(PCGMethodNode selector: #yourself)
    bodyBlock: [ :body | body << #self asPCGNode returnIt ];
    protocol: #accessing;
    yourself
```

You can easily get the code required to generate `Object >> #yourself` via PCG DSL and tweak it.