

# Plaintextgen

Klemens Schindler and Eugen Schindler

Text generation using the MPS templates mechanism

## “Disclaimer” 😊

Anything that we say is our own personal opinion and in no way represents our employer or any other institutions.

# MPS textgen aspect

- Simple model-to-text scenarios with small cognitive gap.

```
text gen component for concept IfStatement {
(node)->void {
    append \n;
    indent buffer;
    append {if (} ${node.condition} {} {});
    with indent {
        append ${node.ifTrue};
    }
    append \n {} $list{node.elsifClauses};
    if (node.ifFalseStatement.isNotNull) {
        append { else } ${node.ifFalseStatement};
    }
}
```

Input model fragment:

```
if (true) {
    System.out.println("test");
}
```

```
concept IfStatement extends Statement
    implements IContainsStatementList
               IDontSubstituteByDefault
               IConditional

instance can be root: false
alias: if
short description: <no short description>

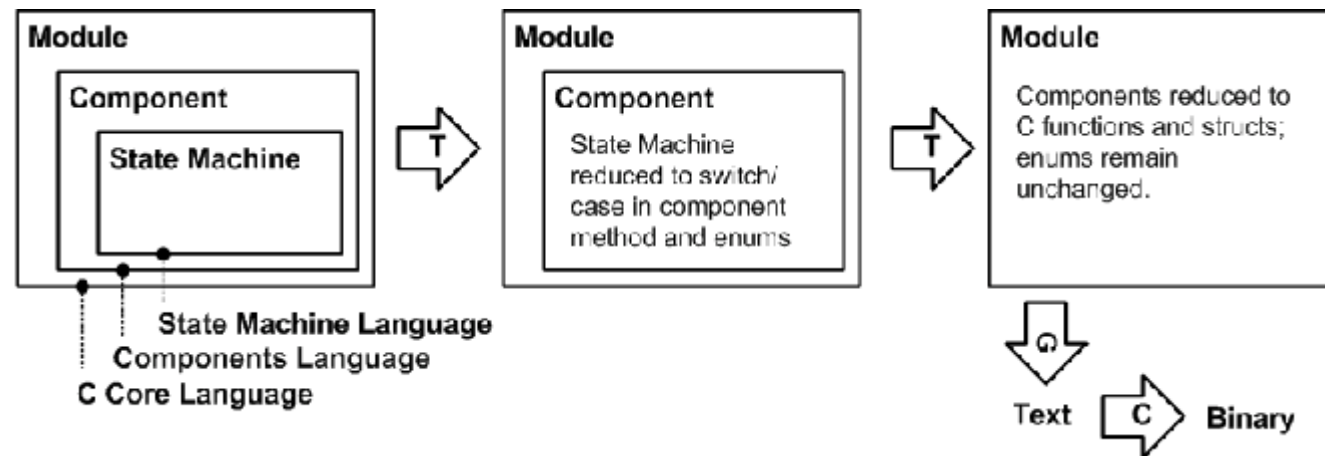
properties:
    forceOneLine      : boolean
    forceMultiLine    : boolean

children:
    condition          : Expression[1]
    ifFalseStatement    : Statement[0..1]
    ifTrue              : StatementList[1]
    elsifClauses        : ElsifClause[0..n]

references:
<< ... >>
```

# MPS textgen aspect

- Often-used approach in MPS for generating to non-fully-supported languages: create a minimum “generation target language” that is easy to textgen.



- Harsh reality: often needed text from various abstraction levels and layers in DSLs, especially when kickstarting and during evolution.

# Plaintextgen history

- Enter [com.dsifoundry.textgen-multilanguage](https://com.dsifoundry.com/textgen-multilanguage): using Sascha Lisson's (now Itemis) richtext to write text and template it, but:
  - No performance
  - Difficult to tweak (richtext not meant for generation extensions and manipulating much text)
- Enter [org.campagnelab.TextOutput](https://org.campagnelab.com/TextOutput): very simple model of words and lines that makes it possible to use the normal reduction rules., but:
  - Tedious to write bigger templates
  - Not enough control over layout (grayspacing, horizontal, vertical)
  - No simple templating from existing text
- Enter [com.dsifoundry.plaintextgen](https://com.dsifoundry.com/plaintextgen):

# Plaintextgen “principles” & features

- Encapsulate textgen aspect & emulate textual-editor appearance (taking editor performance into account)
- Model elements of plaintext allowing more maintainable & understandable text generators
- Performant, so you don't need to switch back to textgen
- Resemble editor language (textual part) → so one less MPS DSL to learn
- Support “classic” text-templating scenarios
- Features:
  - Paste-in support
  - Word-splitting

# Plaintextgen features & usage

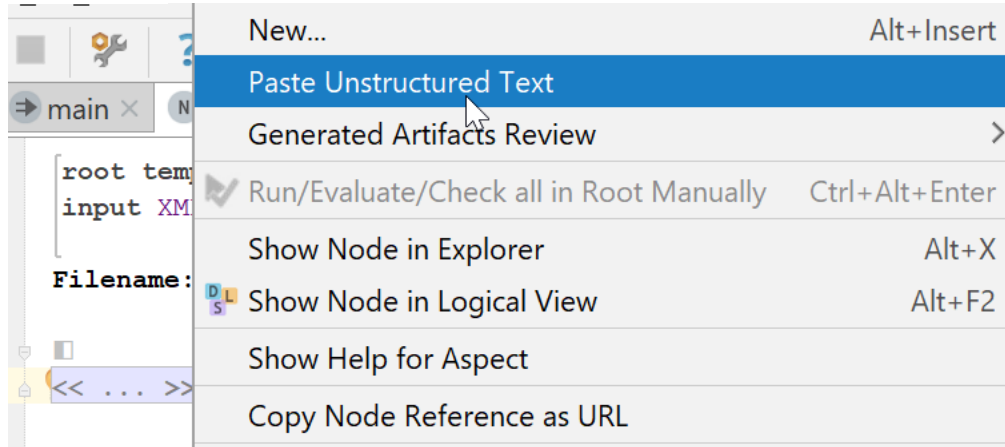
- Features:
  - Paste-in support (Remi Bosman †2018)
  - Powerful layout engine (Remi Bosman †2018)
  - Word-splitting
  - Editor-like wrapping intentions
  - When standard-mechanism not enough → plaintextflow
- General advice: always use only plaintextgen and just skip textgen altogether, examples:
  - For quickly getting text out of any model on any abstraction level
  - For generating to GPLs or horizontal DSLs (e.g. markup languages) that are not yet fully modeled in MPS
- Markus Völter has asked to add plaintextgen to MPS Extensions (maintained by Itemis)
- JetBrains uses plaintextgen (gives migration pull-requests to plaintextgen github repo)

# Typical plaintextgen scenarios

- Create textual template
- Paste into analyzer
- Templatize

```
[root template  
input XMLDocument]
```

Filename: map\_XMLDocument.xml

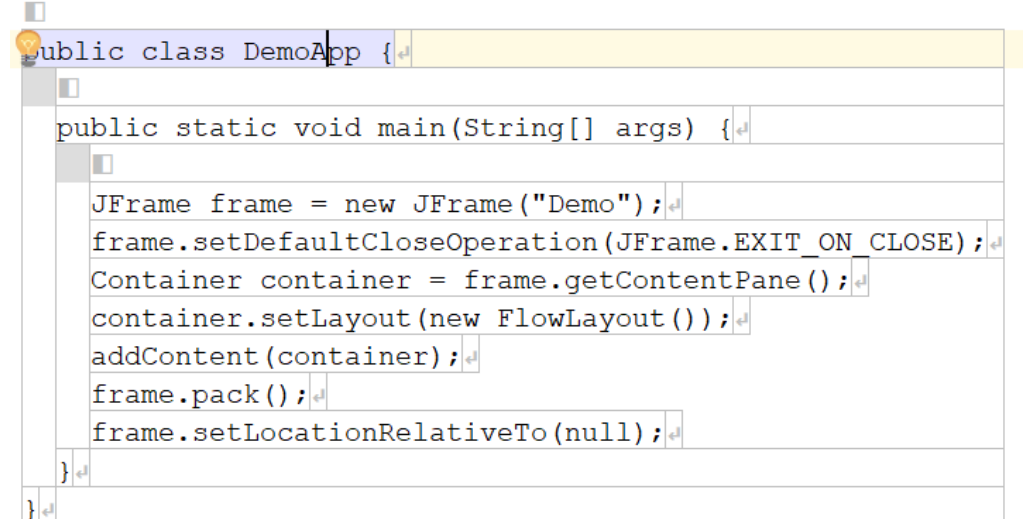


```
[concept XMLDocument --> map_XMLDocument  
inheritors false  
condition <always>  
keep input root default]
```

```
public class DemoApp {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Demo");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        Container container = frame.getContentPane();  
        container.setLayout(new FlowLayout());  
        addContent(container);  
        frame.pack();  
        frame.setLocationRelativeTo(null);  
    }  
}
```

```
[root template  
input XMLDocument]
```

Filename: map\_XMLDocument.xml





Bonus: plaintextflow

# Whitespace-sensitive generation target: reStructuredText

Block quotes are just:

Indented paragraphs,  
and they may nest.

An "empty comment" does not  
consume following blocks.  
(An empty comment is ".." with  
blank lines before and after.)

..

So this block is not "lost",  
despite its indentation.

```
.. list-table:: Frozen Delights!
   :widths: 15 10 30
   :header-rows: 1

   * - Treat
     - Quantity
     - Description
   * - Albatross
     - 2.99
     - On a stick!
   * - Crunchy Frog
     - 1.49
     - If we took the bones out, it wouldn't be
       crunchy, now would it?
   * - Gannet Ripple
     - 1.99
     - On a stick!
```

# Plaintextgen vs Plaintextflow

- Structure is:
  - **VerticalLines/IndentedText**
  - **Line**
  - **Word**
- Similar to an MPS editor or a “div” in HTML
- Newline in **Word** should not be used by design, but behaves as hard wrap (reset to text column 0)
- Newline provided by **Line** is always a soft wrap (respects indentation)
- For introducing newlines, the structure has to be decided up-front
- Indentation symbols may only consist of whitespace and not extra symbols



# Plaintextgen vs Plaintextflow

- Structure is:
  - **Flow**
  - **FlowWord**
  - **HardWrap/SoftWrap**
- Similar to a “stream of words” or a “span” in HTML
- Nesting between concepts is flexible
- No newlines are inserted by structure unless explicitly inserted
- Every Flow can introduce indentation.  
Indentation symbol is duplicated on each **SoftWrap**

# Future of plaintextgen

- Highlights of interesting enhancements in the github issues:
  - Usability enhancements like context assist
  - Indenting nlists
  - Documentation & tutorial
- Will become part of MPS Extensions
- Gets even more important with generator tests
- If you have ideas, bugs, feature requests please submit them as an issue on <https://github.com/DSLFoundry/mps-plaintextgen/issues>
- If you want to help: yes please 😊 It's only a "hobby project"

**Thank you! Any questions left?**