

pattern-writing 0.2

Geraldo Xexéo

Tuesday 26th October, 2021 - 22:27

Contents

1	Introduction	2
2	Options	2
3	Commands	2
3.1	Declaring patterns and anti-patterns	2
3.2	Referring to patterns and anti-patterns	2
3.3	Graph Commands	3
3.4	Pattern description	3
4	Example	3
4.1	Open Doors	3
4.2	Be Helpfull	3
5	Code	3
5.1	Option Processing	3
5.2	Required Packages	5
5.3	Graph Support	5
5.4	Patterns	7
5.5	Anti-Patterns	7
5.6	Pattern Writing Styles	8
5.7	Index making	9

1 Introduction

This is a L^AT_EXstyle to help writing pattern languages [Wik].

It directly supports two formats for describing patterns: Portland style (if-then) and Coplien's style.

2 Options

<code>index</code>	
<code>noindex</code>	These options turn pattern indexing true or false
<code>graph</code>	
<code>nograph</code>	These option turn graph generation true or false

3 Commands

In this package you can:

- declare a pattern;
- describe a pattern in two formats, Portland and Coplien's;
- refer to a pattern;
- declare an anti-pattern;
- refer to an anti-pattern;
- create a graph, and
- configure some parameters.

3.1 Declaring patterns and anti-patterns

<code>pattern</code>	
<code>anti-pattern</code>	<code>\pattern{<pattern-name>}</code>
	<code>\antipattern{<anti-pattern-name>}</code>

Those macros declare a pattern or a anti-pattern and put the on the index if `index` is turned on. A pattern is also put in the graph, if `graph` is turned on.

3.2 Referring to patterns and anti-patterns

<code>patternref</code>	
<code>antipatternref</code>	<code>\pattern[<how-to-print>]pattern-name</code>
	<code>\antipattern[<how-to-print>]antipattern-name</code>

Those commands make a reference (with link) to patterns and anti-patterns, allowing for changes in the name, such as putting it in the plural.

3.3 Graph Commands

`\pstartgraph`
`\pstopgraph`
`\pgetgraph`
`\setPatternGraphLayout`

3.4 Pattern description

`\portland` `\portland{\langle if part \rangle}{\langle then part \rangle}`
`\coplien` `\coplien {\langle problem \rangle}{\langle context \rangle}{\langle forces \rangle}{\langle solution \rangle}`
 `{\langle reasoning \rangle}{\langle resulting context \rangle}`

4 Example

4.1 Open Doors

IF there is a need to help students **THEN** is good to have a open-door police, and **Be Helpfull** .

4.2 Be Helpfull

IF there is a need to help people **THEN** you need to be really helpful

5 Code

5.1 Option Processing

`index` The style accepts two pair of options.
`noindex` `index/noindex` controls if a index will be generated
`graph` `graph/nograph` controls if it will be possible to generate and use a graph
`nograph` The default is `index` and `graph`

```

1 \newif\if@showindex\@showindextrue%
2 \newif\if@showgraph\@showgraphtrue%
3 \newif\if@graphstarted\@graphstartedfalse%
4 %
5 \DeclareOption{index}{\@showindextrue}%
6 \DeclareOption{noindex}{\@showindexfalse}%

```

Open Doors



Be Helpfull

Figure 1: A figure

```

7 \DeclareOption{graph}{\@showgraphtrue}%
8 \DeclareOption{nograph}{\@showgraphfalse}%
9 %
10 \ProcessOptions\relax%

```

5.2 Required Packages

`pattern-writing` requires `xparse` to use `NewDocumentCommand` and other syntax, requires `xcolor` to color anti-patterns, `makeidx` to control the index of patterns and `tikz` and sub-packages to draw the graph

```

11 \RequirePackage{xparse}%
12 \RequirePackage{xcolor}%
13 \if@showindex%
14 \RequirePackage{makeidx}%
15 \fi%
16 \if@showgraph%
17 \RequirePackage{pgf,tikz}%
18 \usetikzlibrary {graphs}%
19 %\usetikzlibrary {graphs.standard}%
20 \usetikzlibrary{graphdrawing}%
21 \usegdlibrary{circular,trees,force,layered}%
22 \fi%
23 %

```

5.3 Graph Support

With we use graph, some settings must be enable

```

24 \if@showgraph%
25 \def\p@filename{graph.tikz}%
26 \def\p@CurrentPattern{ZERO}%
27 %
28 % standard layout for the graph
29 %
30 \def\p@GraphLayout{spring layout, node distance = 80mm}%
31 \NewDocumentCommand{\psetfilename}{m}{%
32 \def\p@filename{#1}%
33 }%

```

`\pstartgraph` This macro starts the processing of declarations to create the graph <https://tex.stack-exchange.com/questions/115932/on-the-basics-of-writing-to-reading-from-auxiliary-files-aux-toc-etc>

```

34 \NewDocumentCommand{\pstartgraph}{}{%
35 \newwrite\p@fileh%
36 \immediate\openout\p@fileh=\p@filename%
37 % tried tikz, problem with accents

```

```

38 \immediate\write\p@fileh{\unexpanded{\resizebox{\textwidth}{!}}\@charlb\unexpanded{\begin{tikzpicture}
39 \unexpanded{[]}\@charlb}%
40 \@graphstartedtrue%
41 }%

```

\pstopgraph This macro stops the processing of declarations to create the graph <https://tex.stackexchange.com/questions/115932/on-the-basics-of-writing-to-reading-from-auxiliary-files-aux-toc-etc>

```

42 \NewDocumentCommand{\pstopgraph}{-}{%
43 \immediate\write\p@fileh{\@charrb%
44 \unexpanded{;\end{tikzpicture}}\@charrb}%
45 \immediate\closeout\p@fileh%
46 \@graphstartedfalse%
47 }%

```

\setPatternGraphLayout This macro is used to set tikz commands for the graph layout. Please check tikz for possible values.

```

48 \NewDocumentCommand{\setPatternGraphLayout}{m}{%
49 \def\p@GraphLayout{#1}%
50 }%

```

\pgetgraph

```

51 \NewDocumentCommand{\pgetgraph}{-}{%
52 \IfFileExists{\p@filename}%
53 {\input{\p@filename}}%
54 {}%
55 }%

56 \NewDocumentCommand{\pnode}{m}{%
57 \if@graphstarted
58 \immediate\write\p@fileh{"#1";}%
59 \def\p@CurrentPattern{#1}%
60 \fi
61 }%

62 \NewDocumentCommand{\pedge}{om}{%
63 \if@graphstarted
64 \IfNoValueTF{#1}
65 {\immediate\write\p@fileh{"\p@CurrentPattern" -- "#2";}%}
66 {\immediate\write\p@fileh{"#1" -- "#2";}%}
67 \fi
68 }%

69 \else % Nada
70 \def\p@filename{graph.tikz}%
71 \def\p@currentPattern{ZERO}%
72 \NewDocumentCommand{\psetfilename}{m}{-}%
73 \NewDocumentCommand{\pstartgraph}{-}{-}%
74 \NewDocumentCommand{\pstopgraph}{-}{-}%

```

```

75 \NewDocumentCommand{\pgetgraph}{-}{\includegraphics[width=\textwidth]{example-
    image-a}}%
76 \NewDocumentCommand{\psetcurrentpattern}{m}{%
77 }%
78 \NewDocumentCommand{\pgetcurrentpattern}{-}{%
79 }%
80 \NewDocumentCommand{\pnode}{m}{-}%
81 \NewDocumentCommand{\pedge}{om}{-}%
82 \NewDocumentCommand{\setPatternGraphLayout}{m}{-}%
83 \fi

```

5.4 Patterns

\pattern This macro declares a pattern (as a subsection).

```

84 \NewDocumentCommand{\pattern}{m}{\subsection{#1}\label{sec:#1}%
85   \if@showindex%
86   \index{#1|textbf}%
87   \fi%
88   \if@showgraph%
89   \pnode{#1}
90   \fi
91 }%

```

\patternref This macro makes a reference to a pattern, possibly changing the text in its option

```

92 \NewDocumentCommand{\patternref}{om}{%
93   \IfNoValueTF{#1}%
94   {\hyperref[sec:#2]{\textbf{#2}}}%
95   {\hyperref[sec:#2]{\textbf{#1}}}%
96   \if@showindex%
97   \index{#2}%
98   \fi%
99   \if@showgraph%
100   \pedge{#2}
101   \fi
102 }%

```

5.5 Anti-Patterns

\antipattern This macro declares anti-pattern (as red text).

```

103 \NewDocumentCommand{\antipattern}{m}{\textcolor{red}{#1}%
104   \label{sec:#1}%
105   \if@showindex%
106   \index{\textcolor{red}{#1}|textbf}%
107   \fi%
108 }%

```

`\antipatternref` This macro makes a reference to an anti-pattern, also in red text

```

109 \NewDocumentCommand{\antipatternref}{om}{%
110   \IfNoValueTF{#1}%
111   {\hyperref[sec:#2]{\textcolor{red}{\textbf{#2}}}}%
112   {\hyperref[sec:#2]{\textcolor{red}{\textbf{#1}}}}%
113   \if@showindex%
114   \index{\textcolor{red}{#2}}%
115   \fi%
116 }%

```

5.6 Pattern Writing Styles

`\setportlandkeys` `\setportlandkeys{<if-key>}{<then-key>}` sets the values for if-then when writing a Portland-style pattern

```

117 \NewDocumentCommand{\setportlandkeys}{mm}{%
118   \def\portland@ifkey{#1}%
119   \def\portland@thenkey{#2}%
120 }%

```

`portland` `\portland{<if part>}{<then part>}`

```

121 \setportlandkeys{SE}{ENTÃO}
122 \NewDocumentCommand{\portland}{mm}{%
123 \textbf{\portland@ifkey~} #1 \textbf{\portland@thenkey~} #2}%

```

`\setcoplienkeys` `\setportlandkeys{<problem>}{<context>}{<forces>}{<solution>}{<reasoning>}{<resultingcontext>}` sets the values for if-then when writing a Portland-style pattern

```

124 \NewDocumentCommand{\setcoplienkeys}{mmmmmm}{%
125   \def\coplien@key@prob{#1}%
126   \def\coplien@key@cont{#2}%
127   \def\coplien@key@forc{#3}%
128   \def\coplien@key@solu{#4}%
129   \def\coplien@key@reas{#5}%
130   \def\coplien@key@cnrs{#6}%
131 }%
132 \setcoplienkeys{\textbf{Problema:}}{\textbf{Contexto:}}{\textbf{Forças:}}{\textbf{Solução:}}{\textbf{Raciocínio:}}{\textbf{Contexto Resultante:}}

```

`\coplien` `\coplien{<problem>}{<context>}{<forces>}{<solution>}{<reasoning>}{<resultingcontext>}`

Writes a pattern in Coplien format

```

133 \NewDocumentCommand{\coplien}{mmmmmm}{%
134   \textbf{\coplien@key@prob:} #1 \par%
135   \textbf{\coplien@key@cont:} #2 \par%
136   \textbf{\coplien@key@forc:} #3 \par%
137   \textbf{\coplien@key@solu:} #4 \par%

```



```

138     \textbf{\coplien@key@reas}: #5 \par%
139     \textbf{\coplien@key@cnrs}: #6 \par%
140 }%

```

5.7 Index making

```

141 \if@showindex%
142 \makeindex%
143 \fi%

```

References

[Wik] C2 Wiki. *Pattern Definition Thread*. URL: <https://wiki.c2.com/?PatternDefinitionThread> (visited on 10/22/2021).