

The er-oo library*

Claudio Fiandrino[†]

January 23, 2013

1 Introduction

The idea for the library came out while attempting to provide an exhaustive answer to [this question](#) in [TeX.SX](#).

Here is a code with a minimal working example showing how the library should be used:

```
\documentclass{article}

\usepackage{tikz}
\usetikzlibrary{er-oo}

\begin{document}
\begin{tikzpicture}[node distance=2.75cm]
% new objects
\pgfoonew \myentity=new entity()
\pgfoonew \myrel=new relationship()
\pgfoonew \myattr=new attribute()

% entity tool
\myentity.set and draw(tool,Tool,1,0) % this stands for:
% \myentity.set label(tool)
% \myentity.text(Tool)
% \myentity.draw(1,0)
\myattr.set and place(tool-id,\underline{ID},left of=tool) % similarly:
% \myattr.set label(tool-id)
% \myattr.text(\underline{ID})
% \myattr.place(left of=tool)
\myattr.set and place(tool-name,Name,right of=tool)
\myentity.multi connect(tool,{tool-id,tool-name})

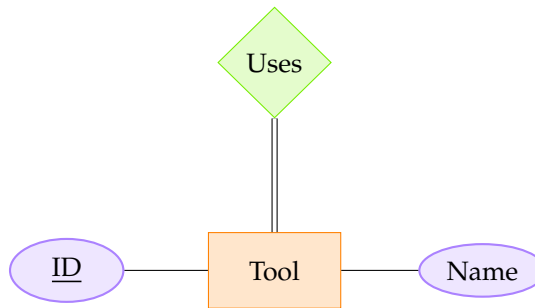
% relation
\myrel.set and place(rel,Uses,above of=tool)
\myrel.total relation(rel,tool)

\end{tikzpicture}
\end{document}
```

*This document corresponds to er-oo 0.2, dated 19/01/2013.

[†]e-mail: claudio dot fiandrino at gmail dot com

The result:



2 Using the library

To use the library you could download the `.dtx` file and compile it with `pdflatex`. This will generate the file `tikzlibraryer-oo.code.tex` which actually is the library (it could also be downloaded directly). Then, the suggested procedure is to install it in your personal tree of your distribution, but it is also possible to copy `tikzlibraryer-oo.code.tex` directly in the directory of your main `.tex` file.

3 Classes and methods

In this section are presented the classes and methods that the library provides. Most of methods are common to all classes, while few of them are specific per class.

There are three classes:

- entity;
- relationship;
- attribute.

Each class has its own constructor method, so to instantiate new objects use:

```
\pgfoonew \myobject=new <class>()
```

where `<class>` is one of the three mentioned above.

3.1 Methods common to all classes

At first are shown methods useful to customize objects.

text

This method sets the text of the object. Usage:

```
\myobject.text(some text)
```

set border color

This method sets the border color of the object. Usage:
`\myobject.set border color(color)`

set fill color

This method sets the background color of the object. Usage:
`\myobject.set fill color(color)`

set text color

This method sets the text color of the object. Usage:
`\myobject.set text color(color)`

set label

This method sets a label for the object. The label could be used subsequently to connect objects. Usage:
`\myobject.set label(label)`

set width

This method defines the width of the object. Usage:
`\myobject.set width(width)`

set height

This method defines the height of the object. Usage:
`\myobject.set height(height)`

The methods lists untill now should be provided *before* actually deploying an object. Object deployment could be realized by means of the following methods.

draw

This method sets the position, in terms of coordinates, where the object is located. Usage:
`\myobject.draw(x,y)`
Example: `\myobject.draw(0,0)`

place

This method sets the position, in terms of the position of another object, where the object is located. Usage:
`\myobject.place(location)`
Example: `\myobject.place(above of=x)` Notice that x is the label of the other object.

set and draw

This method is a composed method: it set the label, the text and the position of an object. Usage:
`\myobject.set and draw(label,text,x,y)`

set and place

This method, similar to the previous one sets the label, the text and the position of an object. Usage:
`\myobject.set and place(label,text,position)`

Now the methods that draw the connections. Connections are depicted by means of the object labels, thus it is important to provide them always.

connect

This method connect one object with another one. Usage:
`\myobject.connect(source,destination)`

multi connect

This method connect one object with more than one other object. Usage:
`\myobject.multi connect(source,{list of labels})`
Example: `\myobject.multi connect(source,{label a,label b, label c})`

total relation

This method connect one object with another one with a total relation. Usage:
`\myobject.total relation(source,destination)`

3.2 Method for attribute

The attribute class has a method not present in other classes. This because attributes could be of some type:

- normal;
- derived;
- multi;

and each one has its own representation.

set type

This method customizes the type of the attribute object. Usage:
`\myobject.set type(type)`

By default the normal type is used. To use the derived attribute set

```
\myobject.set type(derived attribute)
```

while to use the multi attribute set

```
\myobject.set type(multi attribute)
```

4 Implementation - TikZ Library

```
1 \usepgfmodule{oo}
2 \usetikzlibrary{automata,shapes}
3
4 \definecolor{er-purple}{rgb}{.67,.51,1}
5 \definecolor{er-green}{rgb}{.464,.932,0}
6
7 % Entity Class
8 \pgfooclass{entity}{
9
10 \attribute text;
11 \attribute border color=orange;
12 \attribute fill color=orange!20;
13 \attribute text color=black;
14 \attribute label;
15 \attribute width=1.75cm;
16 \attribute height=1cm;
17
18 \method entity() {
19 }
20
21 \method text(#1) {
22 \pgfooset{text}{#1}
23 }
24
25 \method set border color(#1) {
26 \pgfooset{border color}{#1}
27 }
28
29 \method set fill color(#1) {
30 \pgfooset{fill color}{#1}
31 }
32
33 \method set text color(#1) {
34 \pgfooset{text color}{#1}
35 }
36
37 \method set label(#1) {
38 \pgfooset{label}{#1}
39 }
40
41 \method set width(#1) {
42 \pgfooset{width}{#1}
43 }
44
```

```

45 \method set height(#1) {
46   \pgfooset{height}{#1}
47 }
48
49 \method draw(#1,#2) {
50   \node [rectangle,
51     draw=\pgfoovalueof{border color},
52     fill=\pgfoovalueof{fill color},
53     text=\pgfoovalueof{text color},
54     minimum width=\pgfoovalueof{width},
55     minimum height=\pgfoovalueof{height},
56   ] (\pgfoovalueof{label}) at (#1,#2) {\pgfoovalueof{text}};
57 }
58
59 \method place(#1) {
60   \node [rectangle,
61     draw=\pgfoovalueof{border color},
62     fill=\pgfoovalueof{fill color},
63     text=\pgfoovalueof{text color},
64     minimum width=\pgfoovalueof{width},
65     minimum height=\pgfoovalueof{height},
66   #1
67   ] (\pgfoovalueof{label}) {\pgfoovalueof{text}};
68 }
69
70 \method set and draw(#1,#2,#3,#4) {
71   \pgfoothis.set label(#1)
72   \pgfoothis.text(#2)
73   \pgfoothis.draw(#3,#4)
74 }
75
76 \method set and place(#1,#2,#3) {
77   \pgfoothis.set label(#1)
78   \pgfoothis.text(#2)
79   \pgfoothis.place(#3)
80 }
81
82 \method connect(#1,#2) {
83   \draw[-] (#1)--(#2);
84 }
85
86 \method multi connect(#1,#2) {
87   \foreach \i in {#2}{
88     \draw[-] (#1)--(\i);
89   }
90 }
91
92 \method total relation(#1,#2) {
93   \draw[-,double,double distance=1.5pt] (#1)--(#2);
94 }
95
96 }
97
98 % Relationship Class
99 \pgfooclass{relationship}{
100

```

```

101 \attribute text;
102 \attribute border color=er-green;
103 \attribute fill color=er-green!20;
104 \attribute text color=black;
105 \attribute label;
106 \attribute width=1.5cm;
107 \attribute height=0.75cm;
108
109 \method relationship() {
110 }
111
112 \method text(#1) {
113   \pgfooset{text}{#1}
114 }
115
116 \method set border color(#1) {
117   \pgfooset{border color}{#1}
118 }
119
120 \method set fill color(#1) {
121   \pgfooset{fill color}{#1}
122 }
123
124 \method set text color(#1) {
125   \pgfooset{text color}{#1}
126 }
127
128 \method set label(#1) {
129   \pgfooset{label}{#1}
130 }
131
132 \method set width(#1) {
133   \pgfooset{width}{#1}
134 }
135
136 \method set height(#1) {
137   \pgfooset{height}{#1}
138 }
139
140 \method draw(#1,#2) {
141   \node [diamond,
142     draw=\pgfoovalueof{border color},
143     fill=\pgfoovalueof{fill color},
144     text=\pgfoovalueof{text color},
145     minimum width=\pgfoovalueof{width},
146     minimum height=\pgfoovalueof{height},
147   ] (\pgfoovalueof{label}) at (#1,#2) {\pgfoovalueof{text}};
148 }
149
150 \method place(#1) {
151   \node [diamond,
152     draw=\pgfoovalueof{border color},
153     fill=\pgfoovalueof{fill color},
154     text=\pgfoovalueof{text color},
155     minimum width=\pgfoovalueof{width},
156     minimum height=\pgfoovalueof{height},

```

```

157     #1
158   ] (\pgfoovalueof{label}) {\pgfoovalueof{text}};
159 }
160
161 \method set and draw(#1,#2,#3,#4) {
162   \pgfoothis.set label(#1)
163   \pgfoothis.text(#2)
164   \pgfoothis.draw(#3,#4)
165 }
166
167 \method set and place(#1,#2,#3) {
168   \pgfoothis.set label(#1)
169   \pgfoothis.text(#2)
170   \pgfoothis.place(#3)
171 }
172
173 \method connect(#1,#2) {
174   \draw[-] (#1)--(#2);
175 }
176
177 \method multi connect(#1,#2) {
178   \foreach \i in {#2}{
179     \draw[-] (#1)--(\i);
180   }
181 }
182
183 \method total relation(#1,#2) {
184   \draw[-,double,double distance=1.5pt] (#1)--(#2);
185 }
186
187 }
188
189 % Attribute Class
190 \tikzset{multi attribute/.style={double,double distance=1.5pt}}
191 \tikzset{derived attribute/.style={dashed}}
192 \tikzset{attribute type/.style={thick,#1}}
193
194 \pgfooclass{attribute}{
195
196   \attribute text;
197   \attribute border color=er-purple;
198   \attribute fill color=er-purple!20;
199   \attribute text color=black;
200   \attribute label;
201   \attribute type;
202   \attribute width=1.5cm;
203   \attribute height=0.35cm;
204
205   \method attribute() {
206   }
207
208   \method text(#1) {
209     \pgfooset{text}{#1}
210   }
211
212   \method set border color(#1) {

```



```

213 \pgfooset{border color}{#1}
214 }
215
216 \method set fill color(#1) {
217 \pgfooset{fill color}{#1}
218 }
219
220 \method set text color(#1) {
221 \pgfooset{text color}{#1}
222 }
223
224 \method set label(#1) {
225 \pgfooset{label}{#1}
226 }
227
228 \method set type(#1) {
229 \pgfooset{type}{#1}
230 }
231
232 \method set width(#1) {
233 \pgfooset{width}{#1}
234 }
235
236 \method set height(#1) {
237 \pgfooset{height}{#1}
238 }
239
240 \method draw(#1,#2) {
241 \node [ellipse,
242 attribute type={\pgfoovalueof{type}},
243 draw=\pgfoovalueof{border color},
244 fill=\pgfoovalueof{fill color},
245 text=\pgfoovalueof{text color},
246 minimum width=\pgfoovalueof{width},
247 minimum height=\pgfoovalueof{height},
248 ] (\pgfoovalueof{label}) at (#1,#2) {\pgfoovalueof{text}};
249 }
250
251 \method place(#1) {
252 \node [ellipse,
253 attribute type={\pgfoovalueof{type}},
254 draw=\pgfoovalueof{border color},
255 fill=\pgfoovalueof{fill color},
256 text=\pgfoovalueof{text color},
257 minimum width=\pgfoovalueof{width},
258 minimum height=\pgfoovalueof{height},
259 #1
260 ] (\pgfoovalueof{label}) {\pgfoovalueof{text}};
261 }
262
263 \method set and draw(#1,#2,#3,#4) {
264 \pgfoothis.set label(#1)
265 \pgfoothis.text(#2)
266 \pgfoothis.draw(#3,#4)
267 }
268

```

```

269 \method set and place(#1,#2,#3) {
270   \pgfoothis.set label(#1)
271   \pgfoothis.text(#2)
272   \pgfoothis.place(#3)
273 }
274
275 \method connect(#1,#2) {
276   \draw[-] (#1)--(#2);
277 }
278
279 \method multi connect(#1,#2) {
280   \foreach \i in {#2}{
281     \draw[-] (#1)--(\i);
282   }
283 }
284
285 \method total relation(#1,#2) {
286   \draw[-,double,double distance=1.5pt] (#1)--(#2);
287 }
288
289 }

```

Here is a complete example:

```

290 \documentclass{article}
291
292 \usepackage{tikz}
293 \usetikzlibrary{er-oo}
294
295 \begin{document}
296
297 \begin{tikzpicture}[node distance=2.75cm]
298 % new objects
299 \pgfoonew \myentity=new entity()
300 \pgfoonew \myrel=new relationship()
301 \pgfoonew \myattr=new attribute()
302
303 % entity tool
304 \myentity.set and draw(tool,Tool,1,0)
305 \myattr.set and place(tool-id,\underline{ID},left of=tool)
306 \myattr.set and place(tool-name,Name,right of=tool)
307 \myentity.multi connect(tool,{tool-id,tool-name})
308
309 % relation
310 \myrel.set and place(rel,Uses,above of=tool)
311 \myrel.total relation(rel,tool)
312
313 % entity person with some customizations
314 \myentity.set text color(blue)
315 \myentity.set border color(red)
316 \myentity.set and place(per,Person,above of=rel)
317
318 \myattr.set and place(per-id,\underline{ID},left of=per)
319
320 \myattr.set type(derived attribute)
321 \myattr.set and place(per-age,Age,right of=per)
322
323 \myattr.set type() % to reset the derived attribute style

```

```

324 \myattr.set and place(per-name,Name,above left of=per)
325
326 \myattr.set type(multi attribute)
327 \myattr.set and place(per-phone,Phone,above of=per)
328
329 \myattr.set type() % to reset the multi attribute style
330 \myattr.set and place(per-addr,Address,above right of=per)
331
332 \myattr.set and place(street,Street,above right of=per-addr)
333
334 \myattr.set and place(city,City,right of=per-addr)
335 \myattr.multi connect(per-addr,{street,city})
336
337 \myentity.multi connect(per,{per-id,per-age,per-name,per-phone,per-addr,rel})
338 \end{tikzpicture}
339 \end{document}

```