Literate Programming meets UML

Dr Alun Moon

November 10, 2003

–TUG 2003–

WEB & UML

- Literate programming has many well known advantages, to those that know it.
- WEBs concentrate on documenting the algorithms, what goes on inside a method
- UML concentrates on the framework in which a method exists
- WEBs and UML should complement each other well.

Tools

- METAPOST has all the geometrical tools of METAFONT to layout a diagram
- the linear equations allow diagram elements to be laid out in relation to each other.
- METAPOST has TEX formatting to deal with the text in UML.

This is being wrtten from scratch as a learning excercise in writing macro packages.

Conventions + Design

- One class per Java file
- One Java file per WEB file
- intemediate `.uml` file is used in and analogous manner to index creation.

Tangled or Weaved?

- weaved as they form part of the documentation
- tangled because the material is defined in the order of the web file, but has to be rearranged

Style for supporting TEX

```
\def\private{$-$}
\def\public{$+$}
\def\package{$\phantom{+}$}
\def\protected{$\sim$}

\def\classformatproperties#1{%
    \vbox{\halign{##\hfil\cr #1 }}}

% List macros after Knuth in
% The TeXbook, page 378
\def\leftlist#1{%
    \def\\##1{\relax##1\cr}%
    \vbox{\halign{##\hfil\cr#1}}}

\def\classformatlist#1{\leftlist#1}
```

Class creation

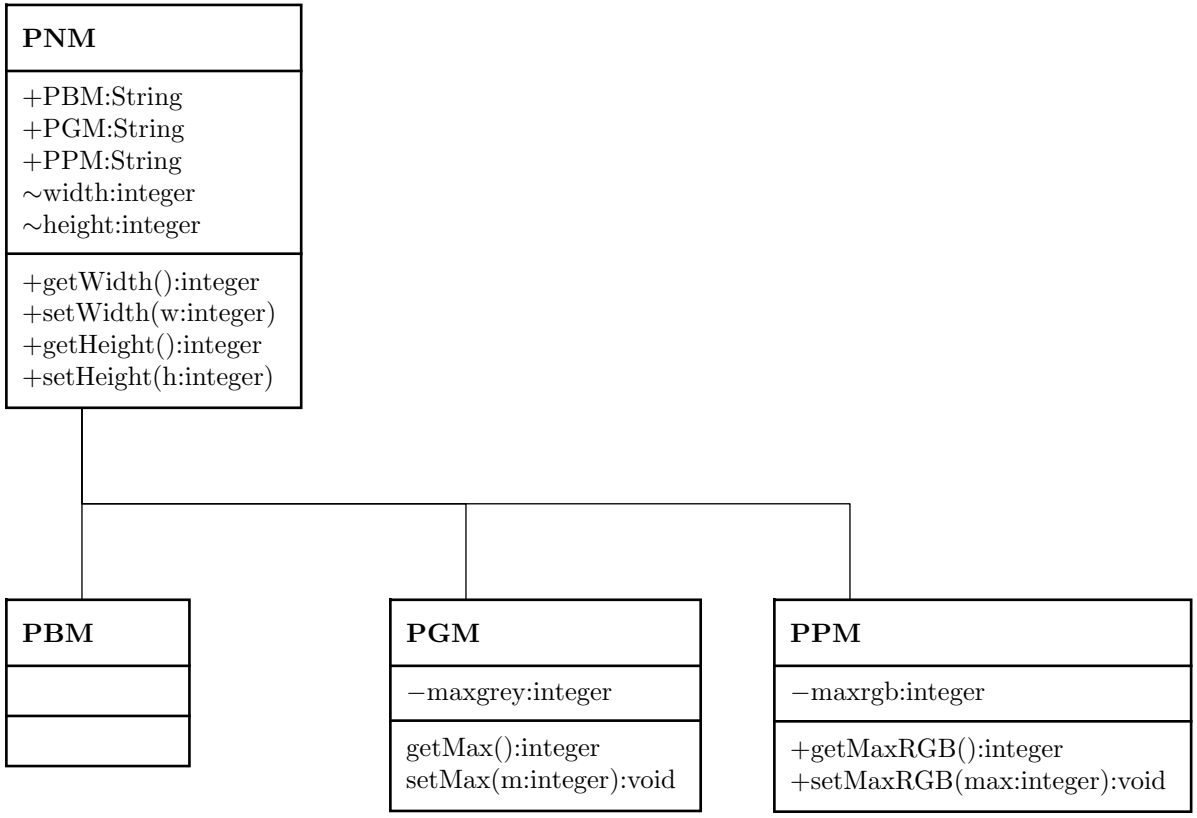
```
vardef class@#(expr title)(expr attributes)(expr operations):=
  save x,y;
  scantokens("pair " & str @# & " top");
  scantokens("pair " & str @# & " bot");
  scantokens("pair " & str @# & " reg");
  scantokens("picture " & str @# & " pic");
  @#pic := nullpicture;
  @#reg + right scaled 1cm = @#top;
  @#top-z0 = @#bot-z6;
  pen ln;
  ln = pensquare scaled 1pt;
  z0 = origin;
  x1-x0 = x3-x2 = x5-x4 = x7-x6 =
  max(width title ,
        width attributes ,
        width operations,
        2cm)+1pc;
  x0 = x2 = x4 = x6;
  y0-y1 = y2-y3 = y4-y5 = y6-y7 = 0;
  y0-y2 = 1.5pc + height title;
  y2-y4 = 1pc   + height attributes;
  y4-y6 = 1pc   + height operations;

  addto @#pic doublepath z0--z1--z7--z6--cycle withpen ln;
  addto @#pic doublepath z2--z3 withpen ln;
  addto @#pic doublepath z4--z5 withpen ln;
  addto @#pic also title shifted (z2+(.5pc,.75pc));
  addto @#pic also attributes shifted (z4+(.5pc,.5pc)-llcorner attributes);
  addto @#pic also operations shifted (z6+(.5pc,.5pc)-llcorner operations);
enddef;
```

Class use

```
beginfig(0)
  class.pnm(btex \bf PNM etex)
  (btex \classformatlist{
        \\{\public PBM:String}
        \\{\public PGM:String}
        \\{\public PPM:String}
        \\{\protected width:integer}
        \\{\protected height:integer}} etex)
  (btex \classformatlist{
        \\{\public getWidth():integer}
        \\{\public setWidth(w:integer)}
        \\{\public getHeight():integer}
        \\{\public setHeight(h:integer)}} etex);

  class.pbm(btex \bf PBM etex)(btex ~ etex)(btex ~ etex);

  class.pgm(btex \bf PGM etex)
  (btex \classformatlist{
        \\{\private maxgrey:integer}} etex)
  (btex \classformatlist{
        \\{getMax():integer}
        \\{setMax(m:integer):void}} etex);

  class.ppm(btex \bf PPM etex)
  (btex \classformatlist{
        \\{\private maxrgb:integer}} etex)
  (btex \classformatlist{
        \\{\public getMaxRGB():integer}
        \\{\public setMaxRGB(max:integer):void}}etex);
```

```
  pnm.reg = origin;
  pnm.bot - pbm.top = (0,1in);
  ppm.reg - pgm.reg = pgm.reg - pbm.reg = (2in,0);

  forsuffixes $=pnm,pbm,pgm,ppm: drawclass$ ; endfor;

  draw pbm.top connect pnm.bot ;
  draw pgm.top connect pnm.bot;
  draw ppm.top connect pnm.bot;
endfig;
```

The class diagram



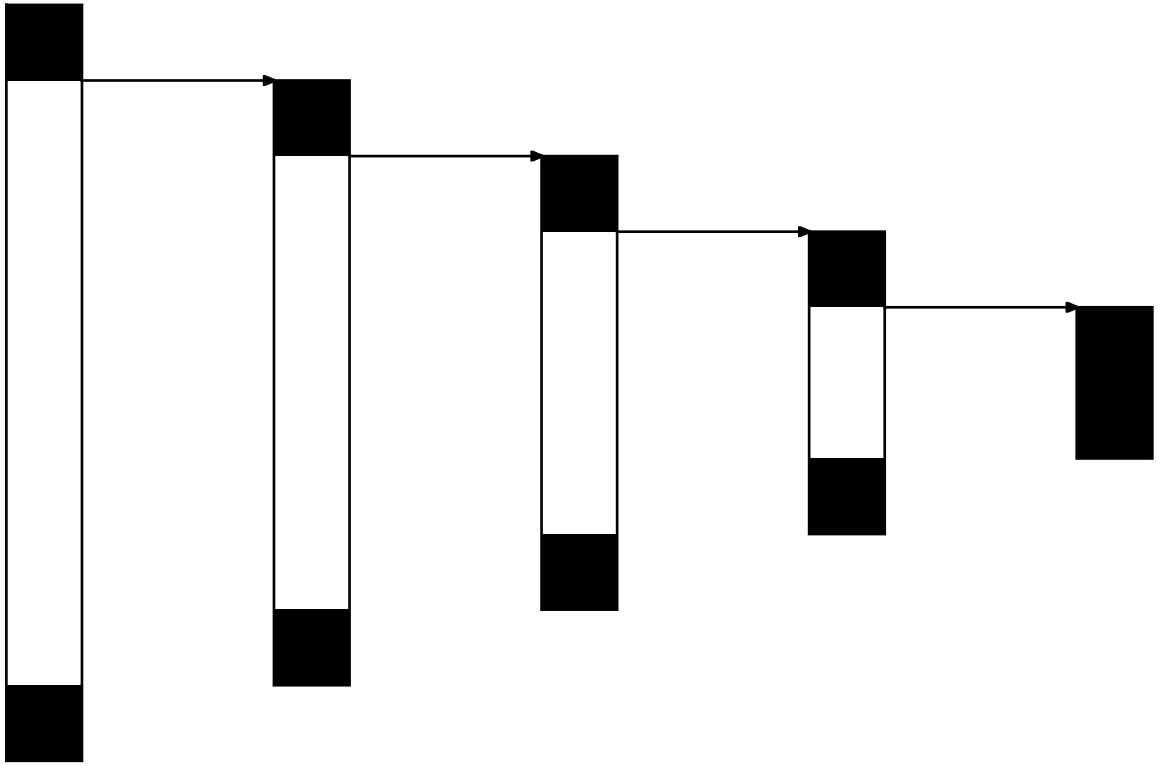
Sequence diagram

```
vardef sequ@#(text call_list) =
  @#.n = .5[@#.nw,@#.ne];
  @#.s = .5[@#.sw,@#.se];
  @#.ne - @#.nw
  = @#.se - @#.sw
  = @#.ce - @#.cw
  = @#.re - @#.rw
  = (seq_width,0);
  @#.nw - @#.cw = @#.rw - @#.sw
  = @#.ne - @#.ce = @#.re - @#.se = (0,seq_width);
  @#.nw - @#.sw = (0,whatever);
  if (length(str call_list) >0):
        @#.ce  + (seq_space,0) = call_list.nw;
        @#.re  + (seq_space,0) = call_list.sw;
  else:
        @#.ce = @#.re;
  fi;
enddef;
```

Use in sequence

```
declaresequence.main;
declaresequence.bezier;
declaresequence.bernstein;
declaresequence.binomial;
declaresequence.fact;

sequ.main(bezier);
sequ.bezier(bernstein);
sequ.bernstein(binomial);
sequ.binomial(fact);
sequ.fact();

main.nw = origin;

beginfig(0)
  pickup pensquare scaled 1pt;
  drawsequence.main;
  drawsequence.bezier;
  drawsequence.bernstein;
  drawsequence.binomial;
  drawsequence.fact;
  drawarrow main.ce--bezier.nw;
  drawarrow bezier.ce--bernstein.nw;
  drawarrow bernstein.ce--binomial.nw;
  drawarrow binomial.ce--fact.nw;
endfig;
```

The diagram



Modifying CWEB

- CWEB produces C++, which is close enough to Java. A web file exists using the `@s` mechanism to modify the syntax
- UML creation can be done largely through TEX macros via an intermediate `.uml` file. Just as indexes are produces to be read in as a set of macros, after sorting and cross-referencing.
- By choosing good macro names and calling conventions, there is a lot that a language such as Perl can do, especially if helpful data is put into comments in the web source and intermediate files
- A simple `sed` script (`sed -e 's/^#/\/\//'`) converts the # line pragmas into line comments. (Can anyone come up with a version of `javac` that can make use of the # line pragmas!)

Web UML meta-tools

- class builder – to collect the attribute and operation lines and write out the TEX/METAPOST class macro.

- sequencer – to arrange the sequences, write out all the sections, declarations, creation, and drawing.

Revised structure

- sequence block would be refered to as $l_2 s_3$
- third sequence block down in the second swim-lane
- less readable METAPOST code
- better for automated generation of complex diagrams

Teaching

- Masters in embedded systems – will be introduced as a possible tool
- Second year undergraduates – suggested as a way to help students think about the design (engineering) of program code, by concentrating on the documentation rather than the coding.