

# umlgraph and the declarative drawing of diagrams

**diomidis spinellis**  
[www.spinellis.gr](http://www.spinellis.gr)

**accu**  
professionalism in programming





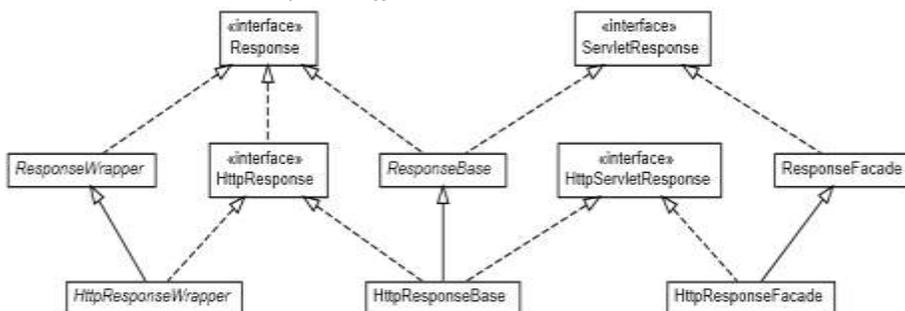




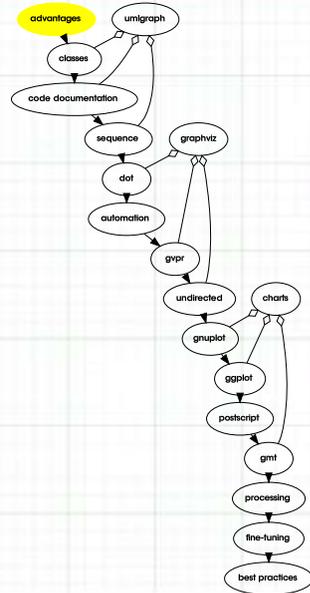
```

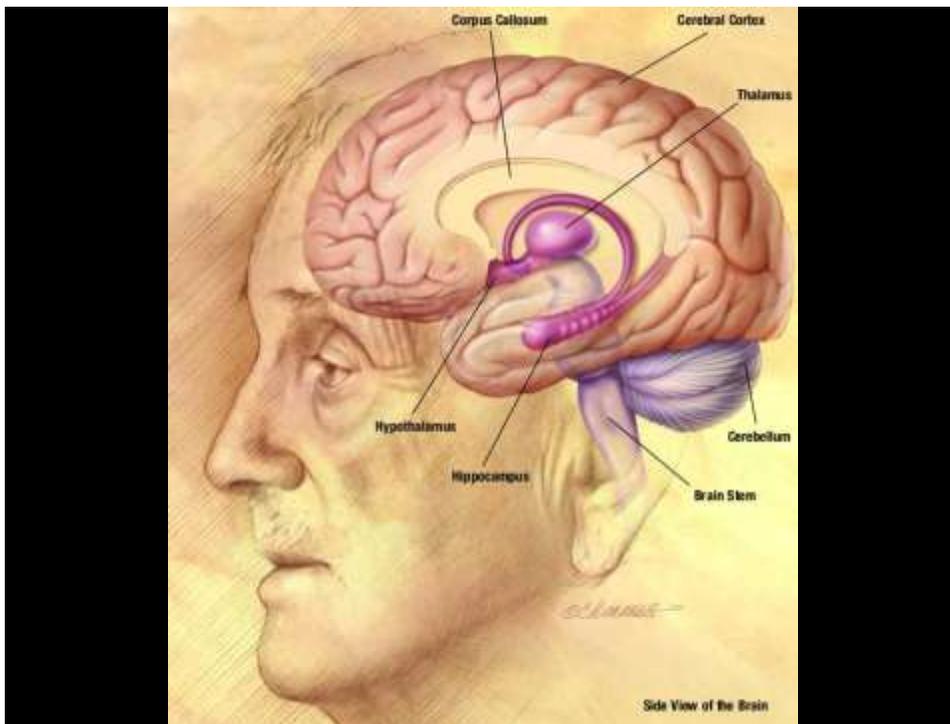
class HttpResponseBase extends ResponseBase
    implements HttpServletResponse {}
abstract class HttpResponseWrapper extends ResponseWrapper
    implements HttpServletResponse {}
class HttpResponseFacade extends ResponseFacade
    implements HttpServletResponse {}
abstract class ResponseWrapper implements Response {}
abstract interface HttpServletResponse extends Response {}
abstract class ResponseBase implements Response, ServletResponse {}
abstract interface HttpServletResponse {}
class ResponseFacade implements ServletResponse {}
abstract interface ServletResponse {}
abstract interface Response {}

```



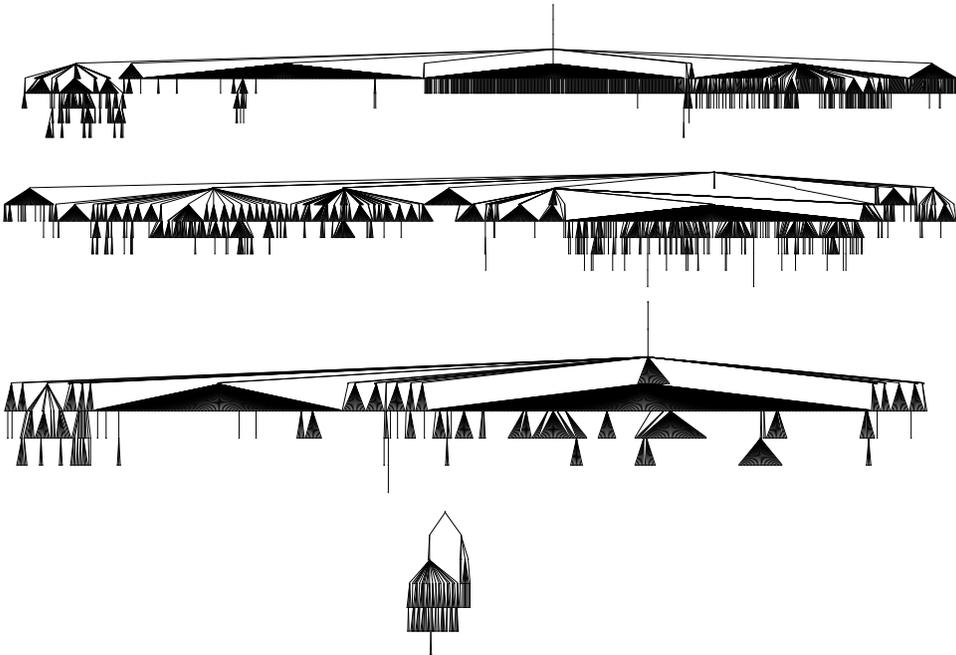
# advantages









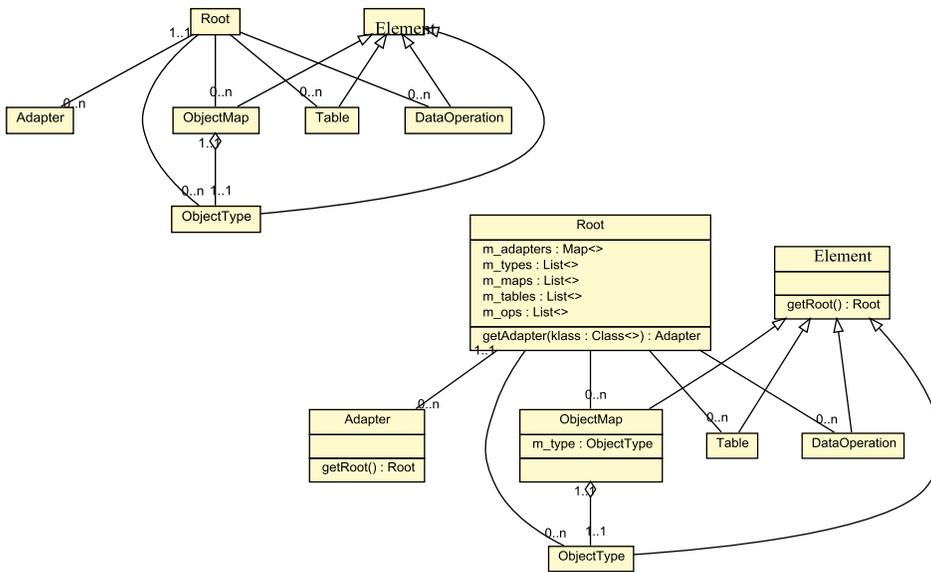


```
C:\dds\src\Research\cscout\refactor>make classdiag.png
grep -h ".*public" *.h | \
egrep -v '(struct|binary_func)' | \
sed 's/class //;s://;s/public //;s/{//;s/ */ /g' | \
awk '{print "class " $1 " extends " $2 " {}"}' >classdiag.java
cmd /c javadoc.exe -docletpath /dds/src/research/umlgraph/lib/UmlGraph.jar -docl
et org.umlgraph.doclet.UmlGraph -private classdiag.java 2>/dev/null
Loading source file classdiag.java...
Constructing Javadoc information...
UmlGraph doclet version 5.4 started
Building general class diagram
25 warnings
mv graph.dot classdiag.dot
dot -Tpng -oclassdiag.png classdiag.dot
rm classdiag.dot

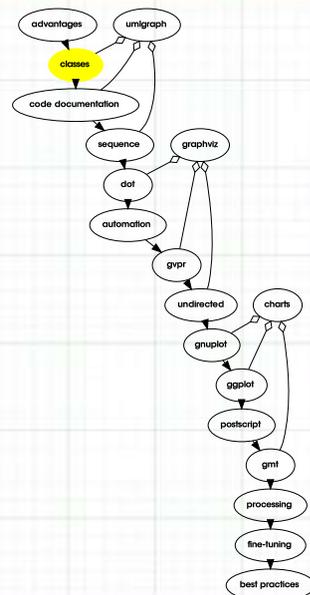
C:\dds\src\Research\cscout\refactor>
```



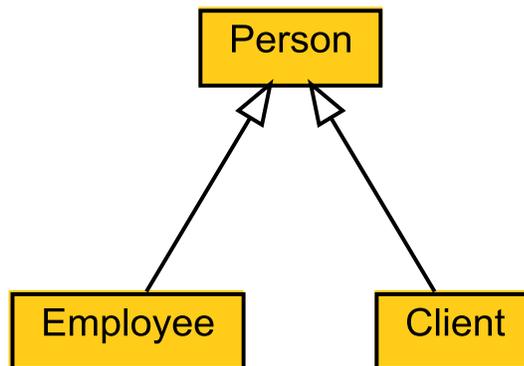




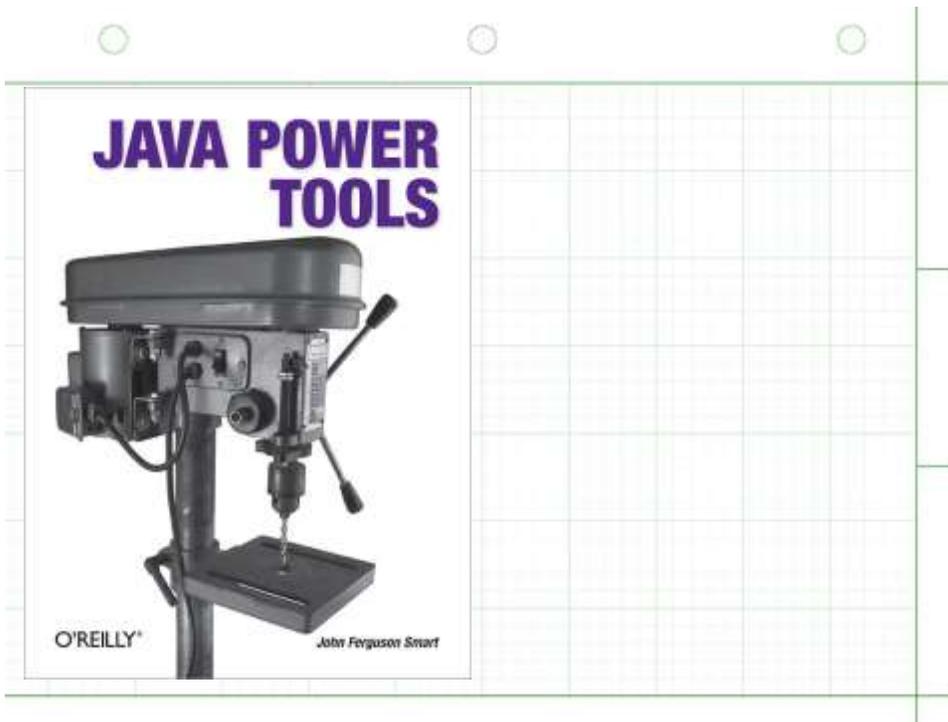
# umlgraph



```
class Person {}  
class Employee extends Person {}  
class Client extends Person {}
```



i thought i'd send you a note saying how much i enjoyed discovering uml graph. i've long fancied a text based way of describing uml diagrams, but never felt sufficiently time-rich to develop anything.

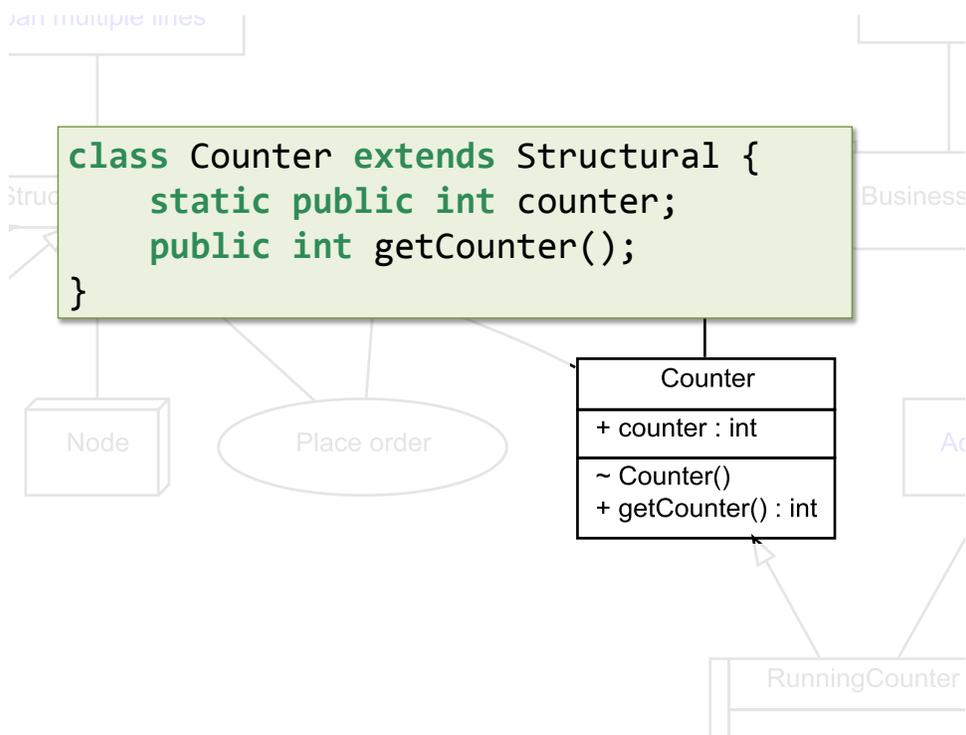
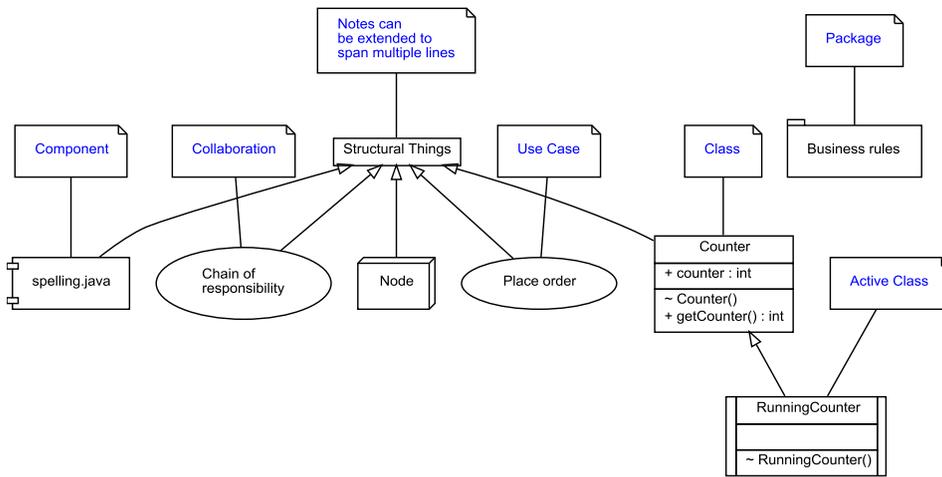


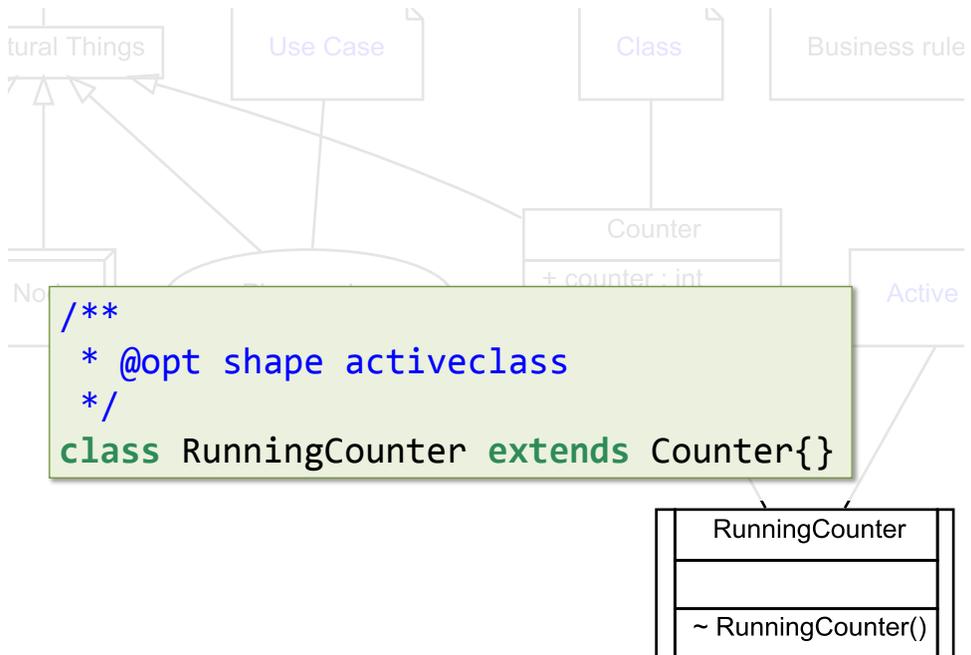
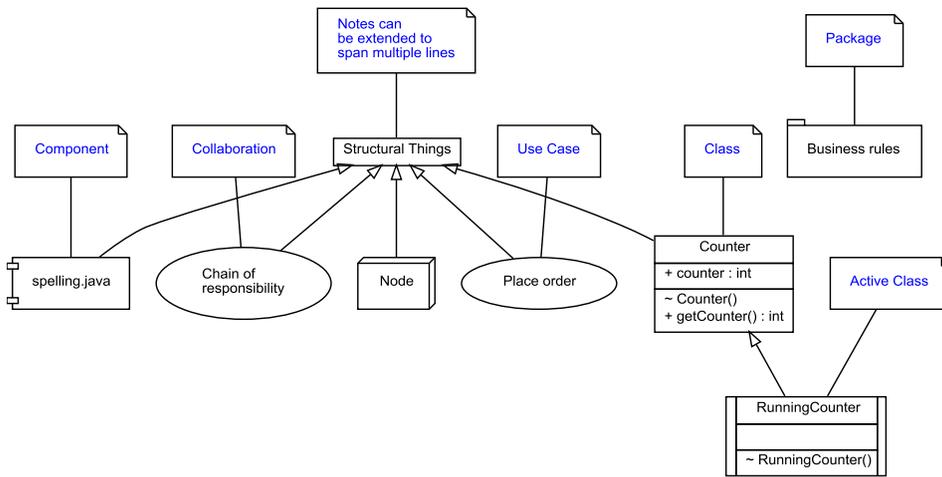
```
echo class HelloWorld {} >Test.java
```

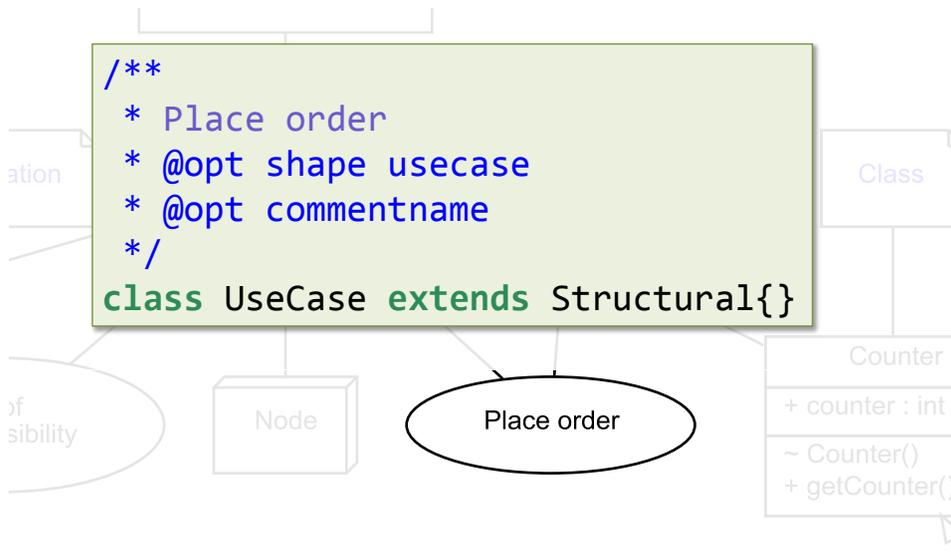
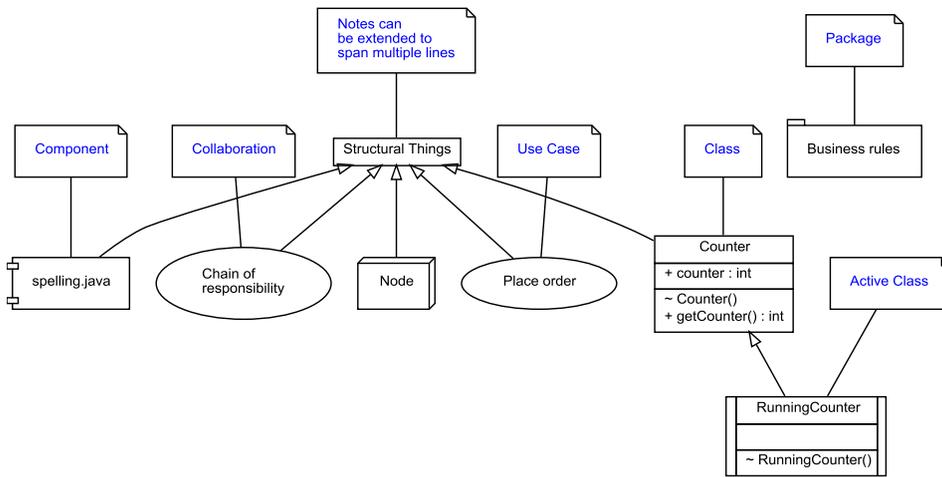
```
umlgraph Test png
```

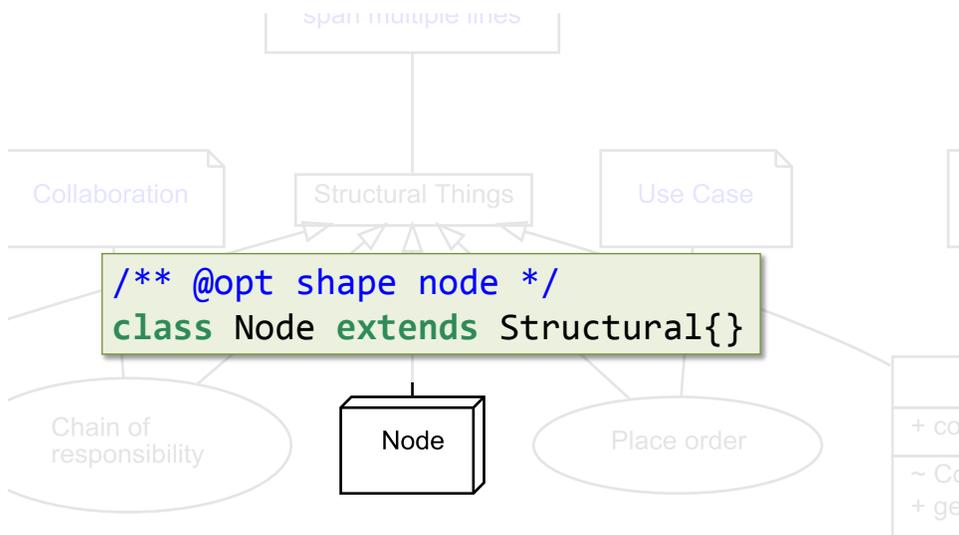
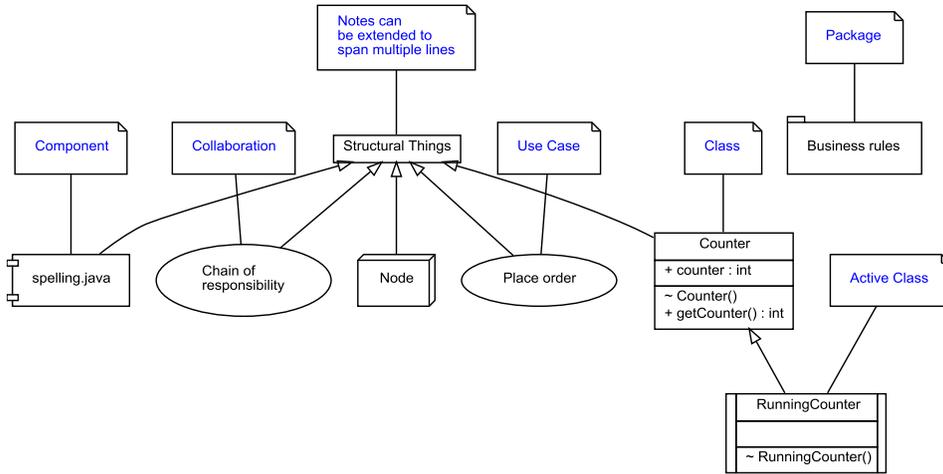
```
# Or one of: bmp canon cmap cmapx cmapx_np dot emf  
emfplus eps fig gd gd2 gif gv imap imap_np ismap jpe  
jpeg jpg metafile pdf plain plain-ext png ps ps2 svg  
svgz tif tiff tk vml vmlz vrml wbmp xdot
```

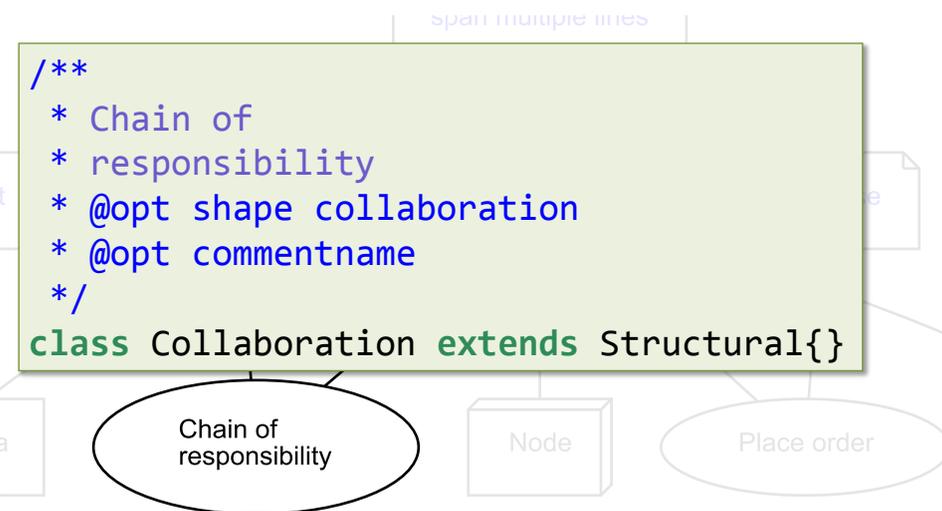
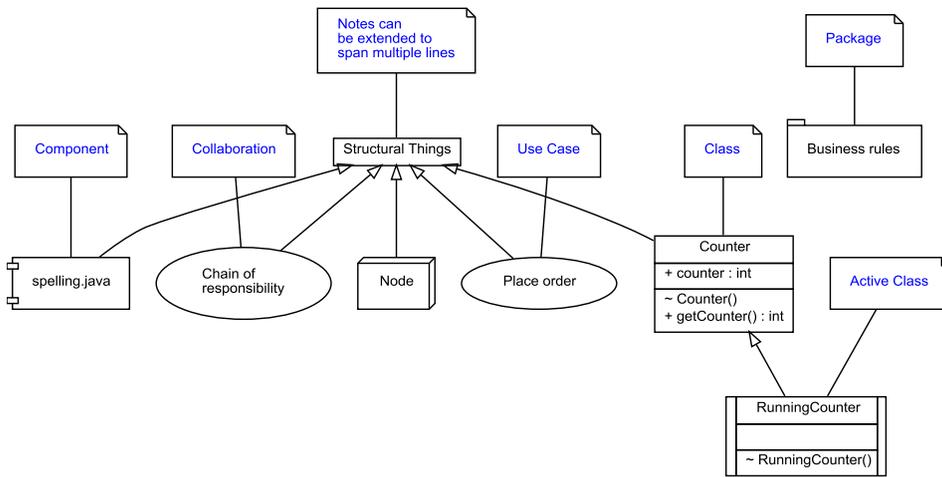
HelloWorld

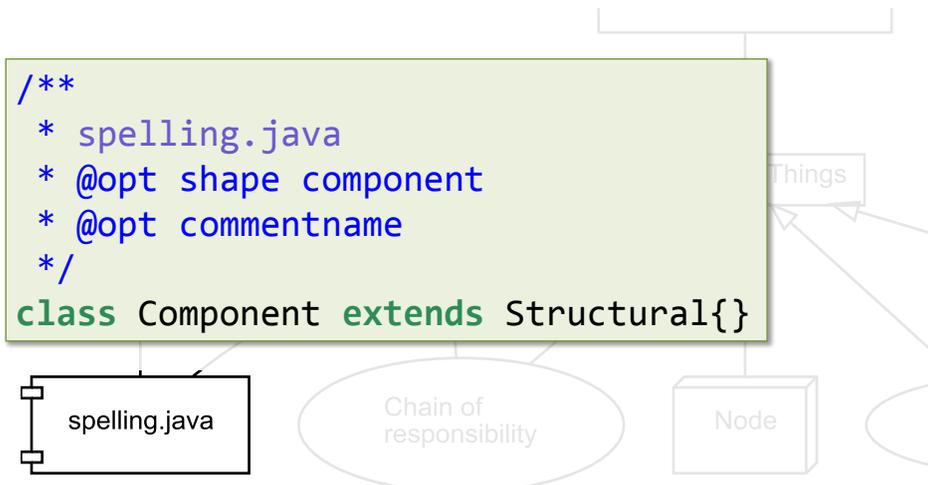
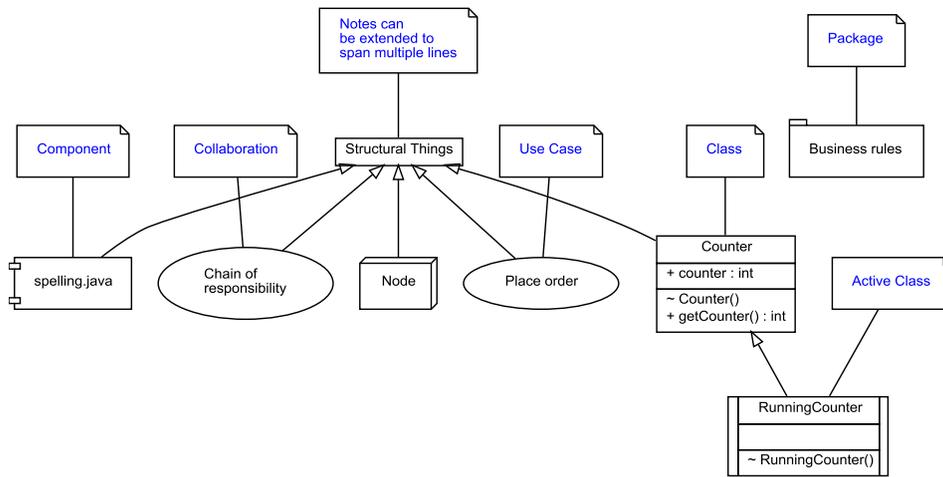


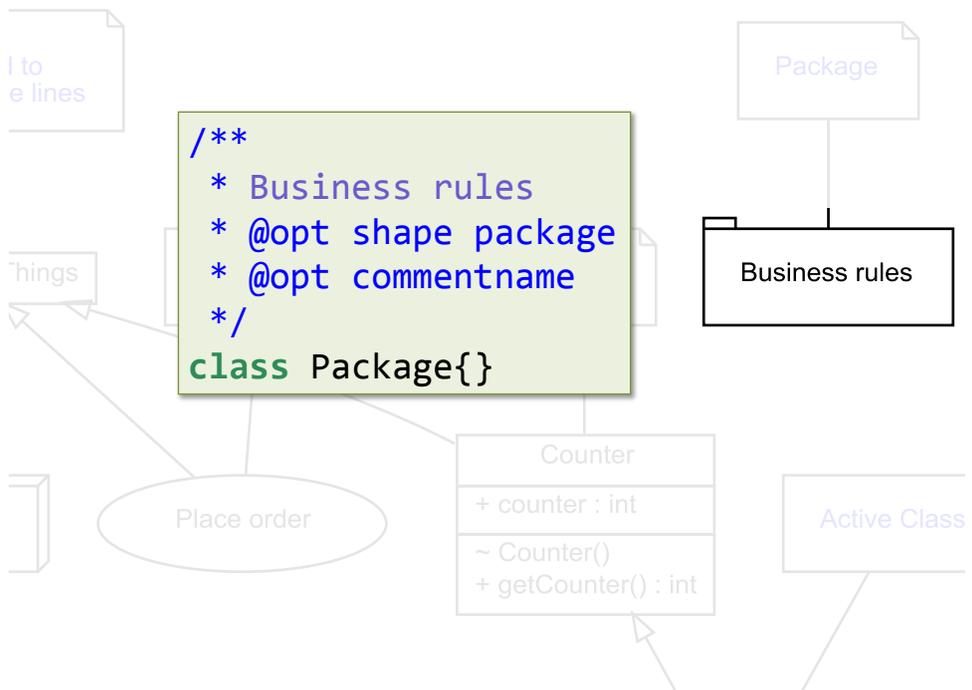
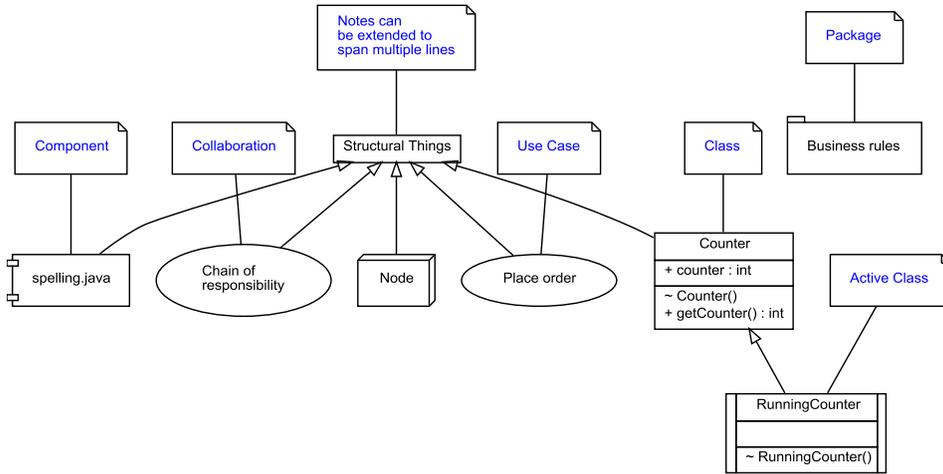


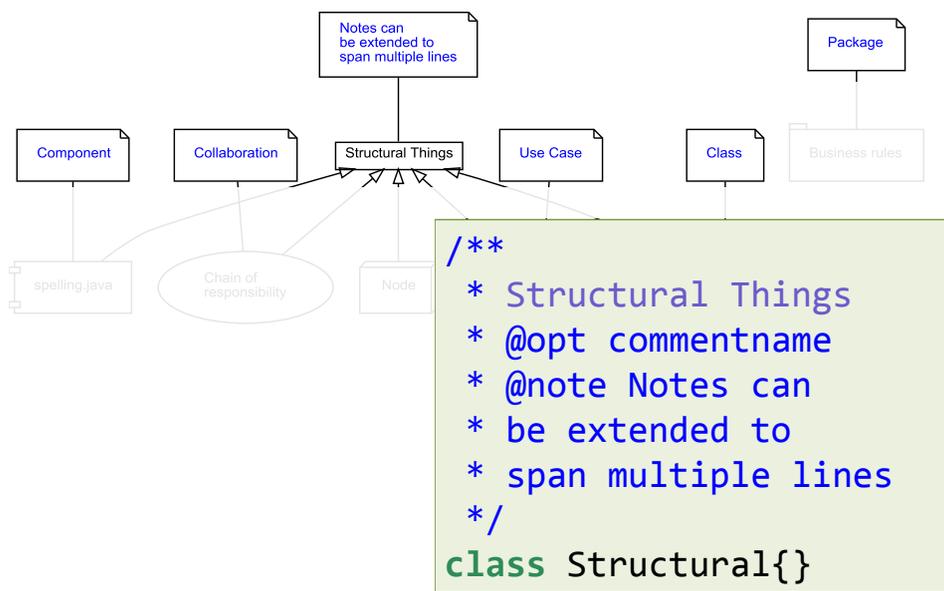
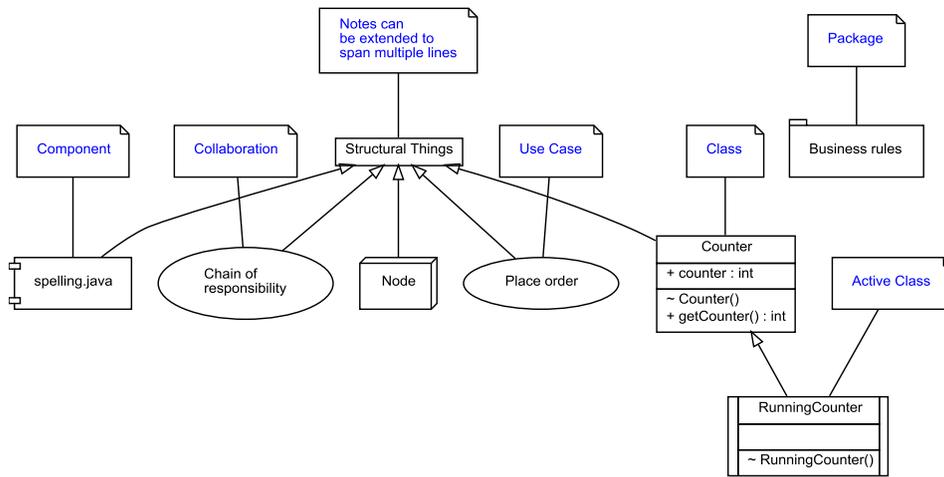








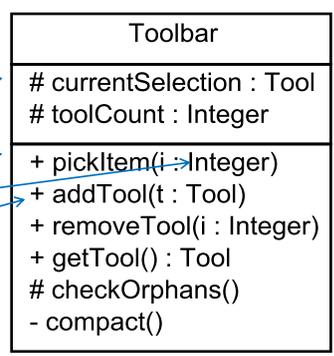




```
/** @hidden */
class Tool {}
```

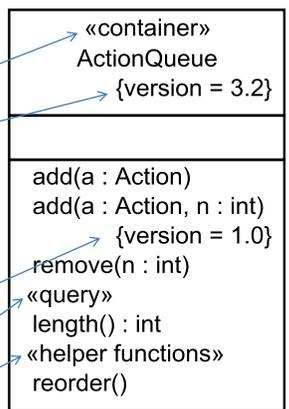
```
/**
 * @opt attributes
 * @opt operations
 * @opt types
 * @opt visibility
 */
```

```
class Toolbar {
    protected Tool currentSelection;
    protected Integer toolCount;
    public void pickItem(Integer i) {}
    public void addTool(Tool t) {}
    public void removeTool(Integer i) {}
    public Tool getTool() {}
    protected void checkOrphans() {}
    private void compact() {}
}
```



```
/**
 * @stereotype container
 * @tagvalue version 3.2
 */
```

```
class ActionQueue {
    void add(Action a) {};
    /** @tagvalue version 1.0 */
    void add(Action a, int n) {};
    void remove(int n) {};
    /** @stereotype query */
    int length() {};
    /** @stereotype "helper functions" */
    void reorder() {};
}
```



# relationships

```
/* Basic categorisations */  
class Asset {}  
class InterestBearingItem {}  
class InsurableItem {}
```

```
/* Asset types */
```

```
/**  
 * @extends InsurableItem  
 * @extends InterestBearingItem  
 */
```

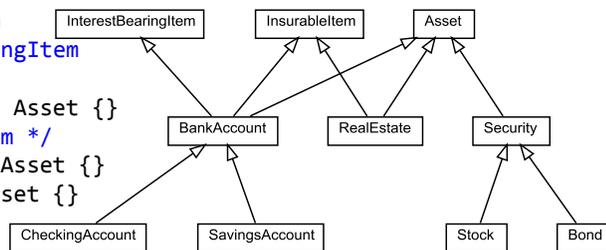
```
class BankAccount extends Asset {}  
/** @extends InsurableItem */  
class RealEstate extends Asset {}  
class Security extends Asset {}
```

```
/* Securities */
```

```
class Stock extends Security {}  
class Bond extends Security {}
```

```
/* Bank accounts */
```

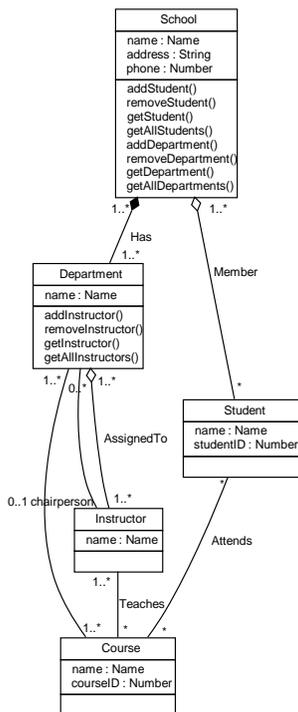
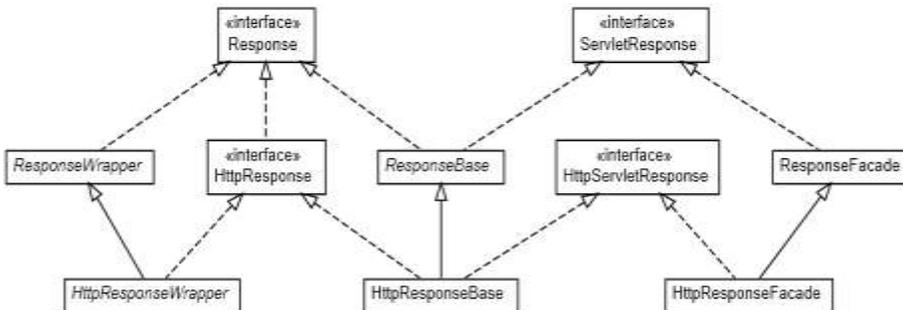
```
class CheckingAccount extends BankAccount {}  
class SavingsAccount extends BankAccount {}
```



```

class HttpResponseBase extends ResponseBase
    implements HttpServletResponse {}
abstract class HttpResponseWrapper extends ResponseWrapper
    implements HttpServletResponse {}
class HttpResponseFacade extends ResponseFacade
    implements HttpServletResponse {}
abstract class ResponseWrapper implements Response {}
abstract interface HttpServletResponse extends Response {}
abstract class ResponseBase implements Response, ServletResponse {}
abstract interface HttpServletResponse {}
class ResponseFacade implements ServletResponse {}
abstract interface ServletResponse {}
abstract interface Response {}

```

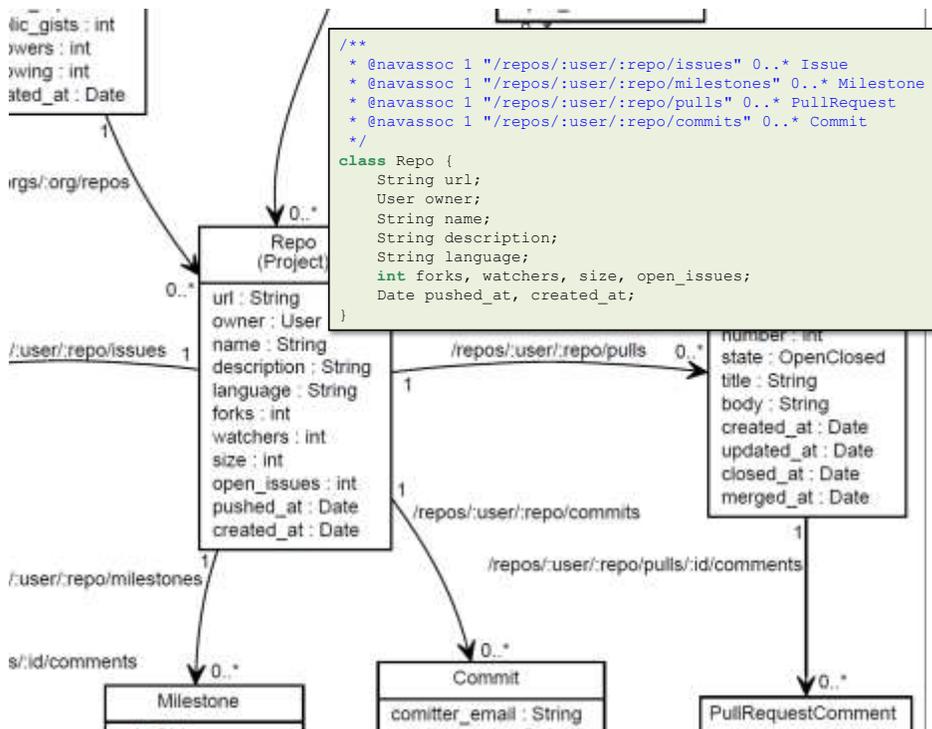


```

/**
 * @has 1..* Member * Student
 * @composed 1..* Has 1..* Department
 */
class School {
    Name name;
    String address;
    Number phone;
    void addStudent() {}
    void removeStudent() {}
    void getStudent() {}
    void getAllStudents() {}
    void addDepartment() {}
    void removeDepartment() {}
    void getDepartment() {}
    void getAllDepartments() {}
}

```





```

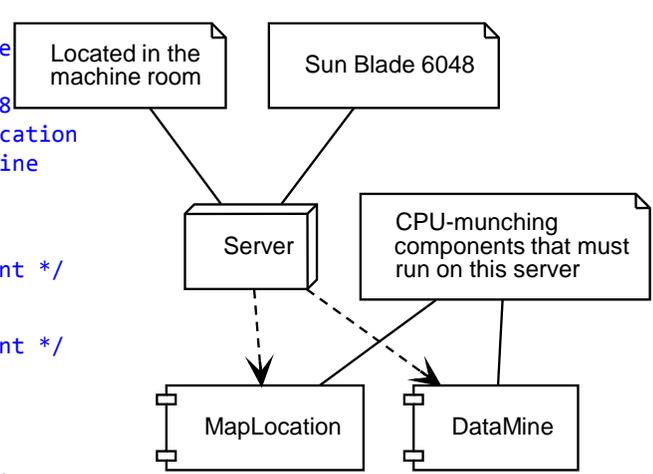
/**
 * @opt shape node
 * @note Located in the machine room
 * @note Sun Blade 6048
 * @depend - - - MapLocation
 * @depend - - - DataMine
 */
class Server{}

/** @opt shape component */
class MapLocation {}

/** @opt shape component */
class DataMine {}

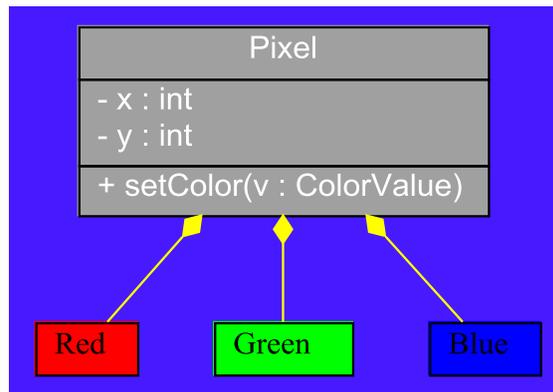
/**
 * CPU-munching
 * components that must
 * run on this server
 * @opt shape note
 * @opt commentname
 * @assoc - - - MapLocation
 * @assoc - - - DataMine
 */
class munchComment {}

```



# options

```
/**
 * @opt edgecolor "yellow"
 * @opt nodefontname "Times"
 * @opt bgcolor ".7 .9 1"
 * @opt nodefillcolor "#a0a0a0"
 * @opt nodefontsize 14
 * @hidden
 */
class UMLOptions{}
/**
 * @opt nodefontname "Arial"
 * @opt nodefontcolor "white"
 * @composed - - - Red
 * @composed - - - Green
 * @composed - - - Blue
 * @opt operations
 * @opt visibility
 * @opt types
 */
class Pixel {
    private int x, y;
    public void setColor(ColorValue v) {}
}
/** @opt nodefillcolor red */
class Red {}
/** @opt nodefillcolor green */
class Green {}
/** @opt nodefillcolor blue */
class Blue {}
```



```

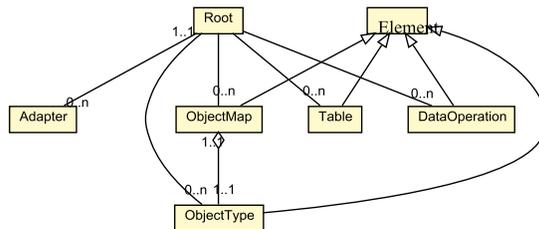
/**
 * @view
 * @opt viewOption1
 * @opt viewOption2
 * ...
 * @match matchtype regularExpression
 * @opt option1.1 [argument]
 * @opt option1.2 [argument]
 * ...
 * @match matchtype regularExpression
 * @opt option2.1 [argument]
 * @opt option2.2 [argument]
 * ...
 /

```

```

/**
 * @view
 */
class Overview {}

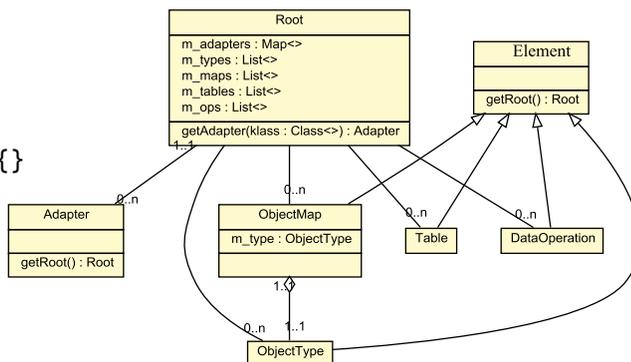
```



```

/**
 * @view
 * @opt attributes
 * @opt operations
 */
class DetailedView {}

```



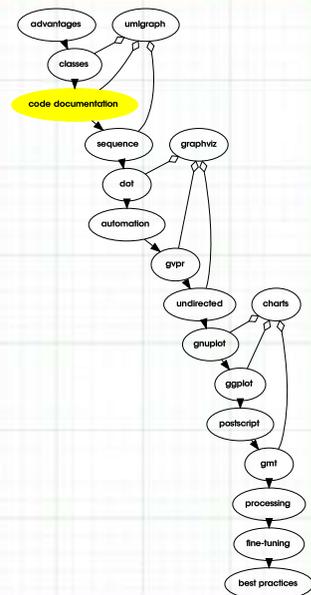




# reverse engineering

“Instead of wasting time designing our software, why don’t we spend this time to hack on the code. Then, we can satisfy our organization’s (or our client’s) software process Gestapo by deriving UML diagrams from the code.”

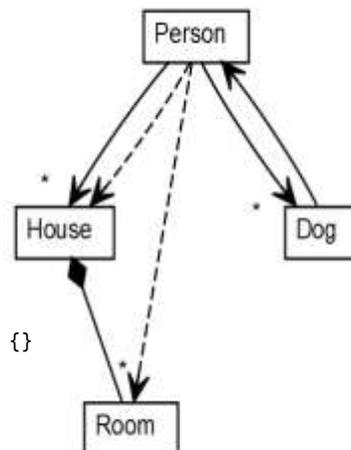
code docs



# relationship inference

- inferdep
- inferrel
  
- collpackages
- inferdepinpackage
- inferdepvis
- inferreltype
- useimports

```
import java.util.List;
import java.util.Map;
/**
 * @opt inferrel
 * @opt collpackages java.util.*
 * @opt inferdep
 * @opt inferdepinpackage
 * @opt hide java.*
 * @hidden
 */
class UMLOptions {}
class Person {
    House[] houses;
    List<Dog> dogs;
    public Room getFavouriteRoom(House house) {}
}
/**
 * @opt inferreltype composed
 */
class House {
    Map<String, Room> nameRoomMap;
}
class Room {}
class Dog {
    Person owner;
}
```



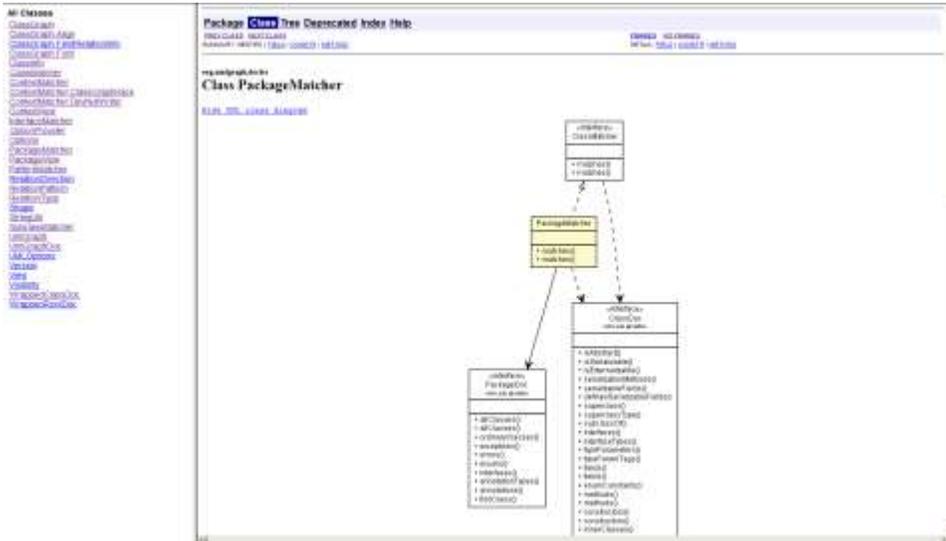
```

<target name="javadocs" depends="compile">
  <javadoc sourcepath="${src}" packagenames="org.umlgraph.doclet.*"
    destdir="${javadoc}" private="true">
    <doclet name="org.umlgraph.doclet.UmlGraphDoc"
      path="${lib}/UmlGraph.jar">
      <param name="-inferrel"/>
      <param name="-inferdep"/>
      <param name="-collapsible"/>
      <param name="-hide" value="java.*"/>
      <param name="-collpackages" value="java.util.*"/>
      <param name="-qualify"/>
      <param name="-postfixpackage"/>
      <param name="-nodefontsize" value="9"/>
      <param name="-nodefontpackagesize" value="7"/>
      <param name="-link"
value="http://docs.oracle.com/javase/7/docs/jdk/api/javadoc/doclet/"/>
      <param name="-link"
value="http://download.oracle.com/javase/7/docs/api/"/>
    </doclet>
  </javadoc>
</target>

```

The screenshot shows the JavaDoc API page for the class `Class PackageMatcher`. The page is organized into several sections:

- Package:** `org.umlgraph.doclet`
- Class PackageMatcher** (indicated by a yellow arrow)
- All Implemented Interfaces:** `ClassMatcher`
- Field Summary:**
  - `packageName`
  - `packageName`
- Constructor Summary:**
  - `ClassPackageMatcher()`
- Method Summary:**
  - `matches(Class[] cls)`: Returns the options for the specified class.
  - `matches(String name)`: Returns the options for the specified class.
- Methods inherited from class java.lang.Object:**
  - `clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()`
- Field Detail:**

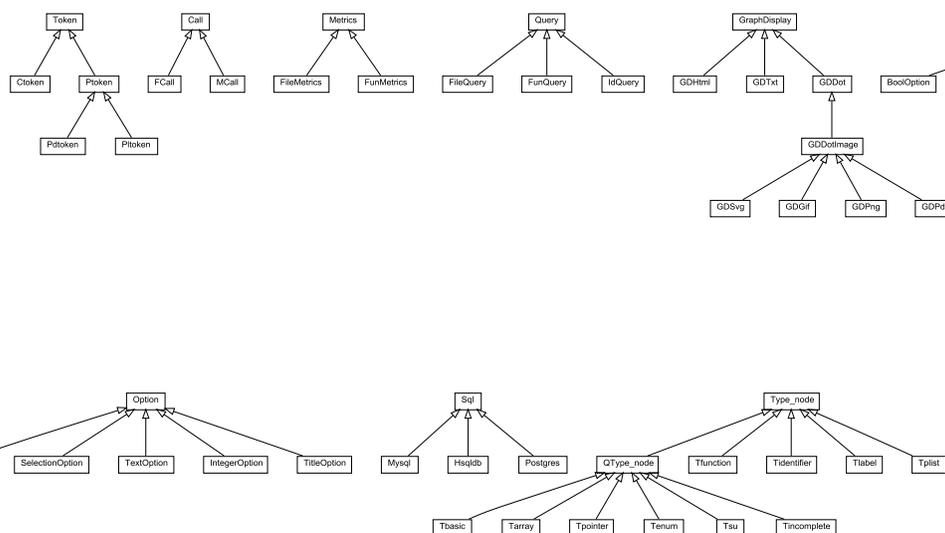


C++

```

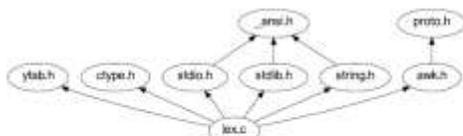
grep -h ".*public" *.h |
egrep -v '(struct|binary_func)' |
sed 's/class //;s/://;s/public //;s/{//;s/ */ /g' |
awk '{print "class " $1 " extends " $2 " {}"}' >diagram.java

```

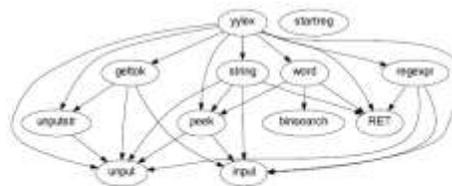




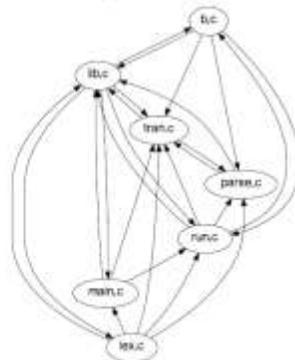
# cscout



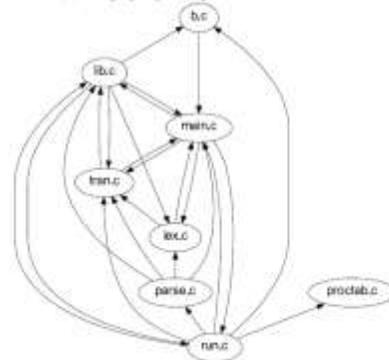
(a) Included files.



(b) Call graph spanning functions and macros.



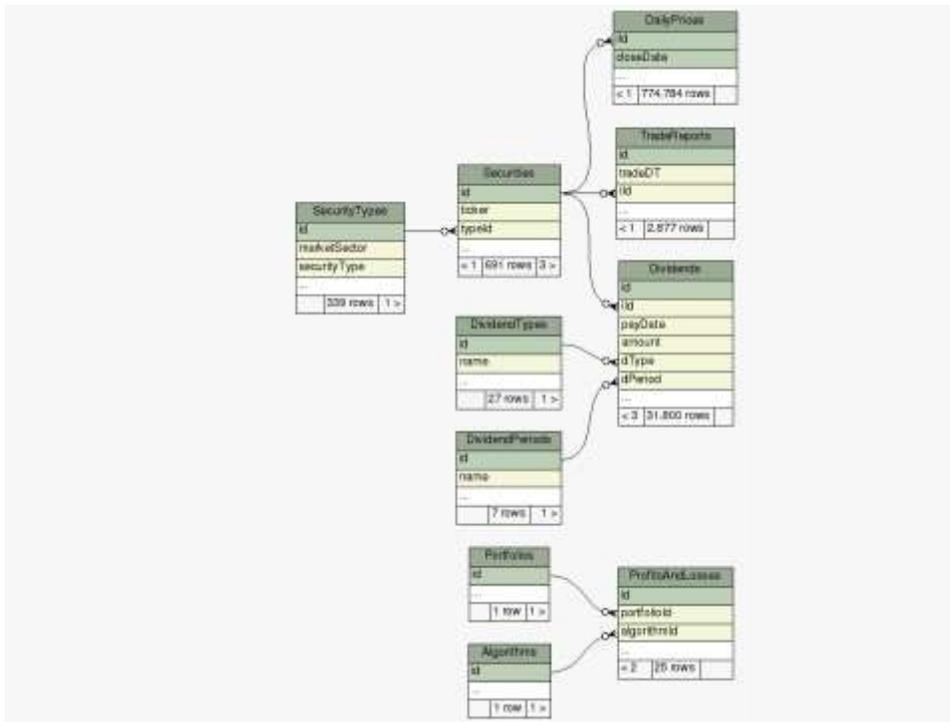
(c) Control dependencies between files.



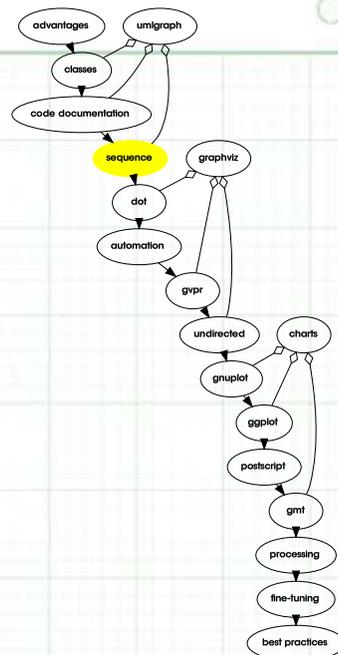
(d) Data dependencies between files.

# schemaSpy

```
java -jar schemaSpy_5.0.0.jar -t mysql \  
-host db.example.us-east-1.rds.amazonaws.com \  
-u admin -p password -o schema \  
-db dbname \  
-dp mysql-connector-java-5.1.19-bin.jar
```



# sequence

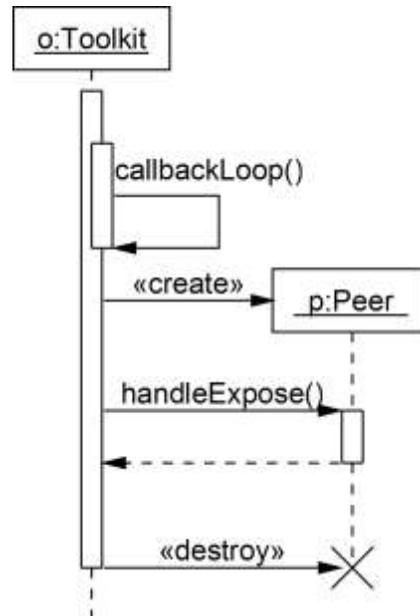


```

.PS
copy "sequence.pic";
# Define the objects
object(O, "o:Toolkit");
placeholder_object(P);
step();
# Message sequences
active(O);
step();
active(O);
message(O,O, "callbackLoop()");
inactive(O);
create_message(O,P, "p:Peer");
message(O,P, "handleExpose()");
active(P);
return_message(P,O, "");
inactive(P);
destroy_message(O,P);
inactive(O);
# Complete the lifelines
step();
complete(O);
.PE

```

pic2plot -Tgif FILENAME.pic >FILENAME.svg



```

.PS
copy "sequence.pic";

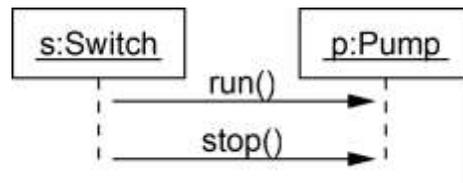
# Define the objects
object(S, "s:Switch");
object(P, "p:Pump");

# Message sequences
message(S,P, "run()");
message(S,P, "stop()");

complete(S);
complete(P);

.PE

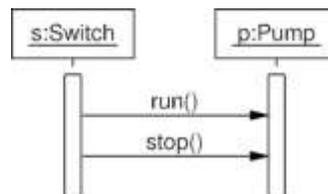
```



```

# Define the objects
object(S, "s:Switch");
object(P, "p:Pump");
step();
active(S);
active(P);

```



```

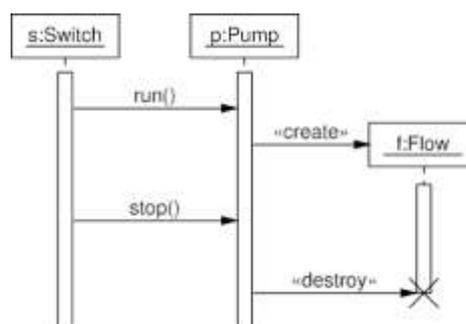
# Message sequences
message(S,P, "run()");
message(S,P, "stop()");
step();
complete(S);
complete(P);

```

```

# Define the objects
object(S, "s:Switch");
object(P, "p:Pump");
placeholder_object(F);
step();
active(S);
active(P);
# Message sequences
message(S,P, "run()");
create_message(P,F, "f:Flow");
active(F);
message(S,P, "stop()");
destroy_message(P,F);
step();
complete(S);
complete(P);

```



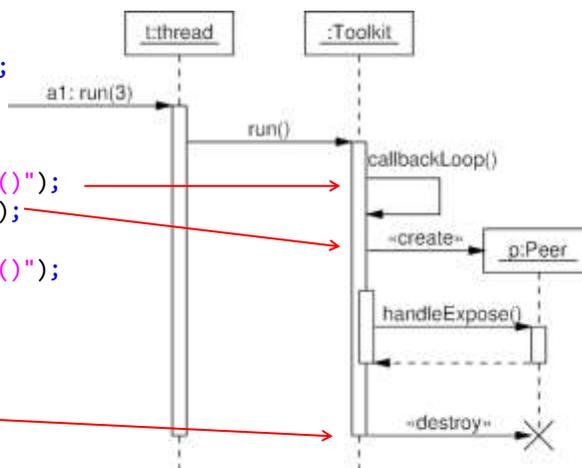
# operations

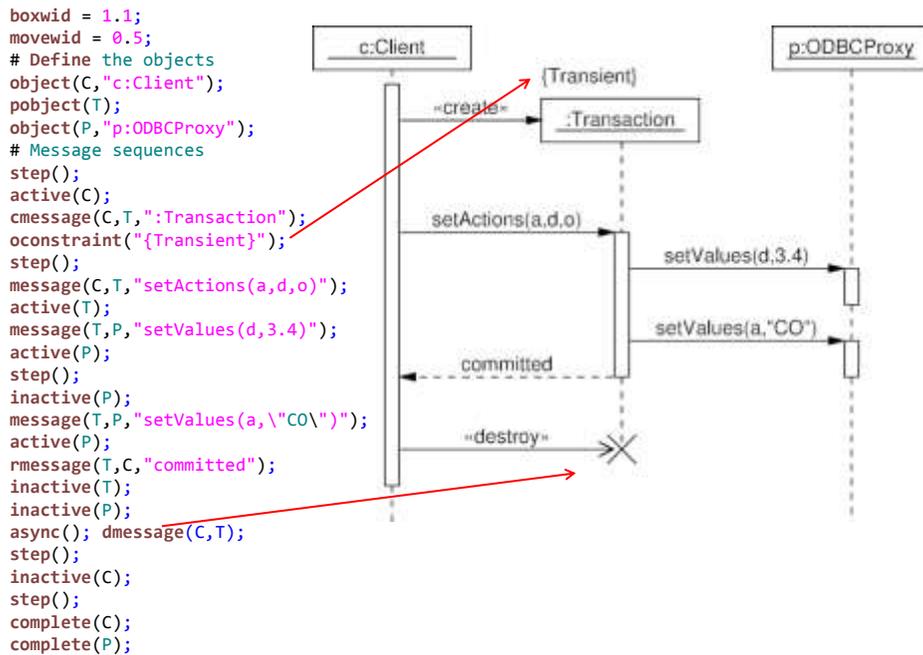
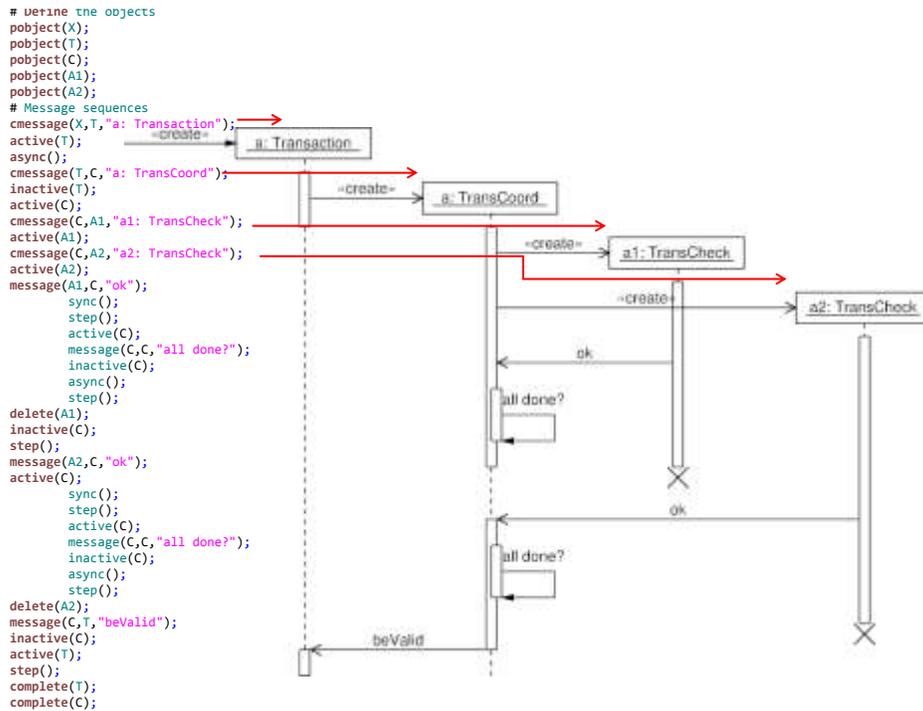
- object
- placeholder
- actor
- complete
- message
- return\_message
- create\_message
- destroy\_message
- active
- inactive
- delete
- lifeline\_constraint
- lconstraint\_below
- object\_constraint
- step
- async
- sync
- begin\_frame
- end\_frame
- comment
- connect to comment

```

pobject(E, "External Messages");
object(T, "t:thread");
object(O, ":Toolkit");
pobject(P);
step();
# Message sequences
message(E,T, "a1: run(3)");
active(T);
message(T,O, "run()");
active(O);
message(O,O, "callbackLoop()");
cmessage(O,P, "p:Peer", " ");
active(O);
message(O,P, "handleExpose()");
active(P);
rmessage(P,O, "");
inactive(P);
inactive(O);
dmessage(O,P);
inactive(T);
inactive(O);
step();
complete(T);
complete(O);

```

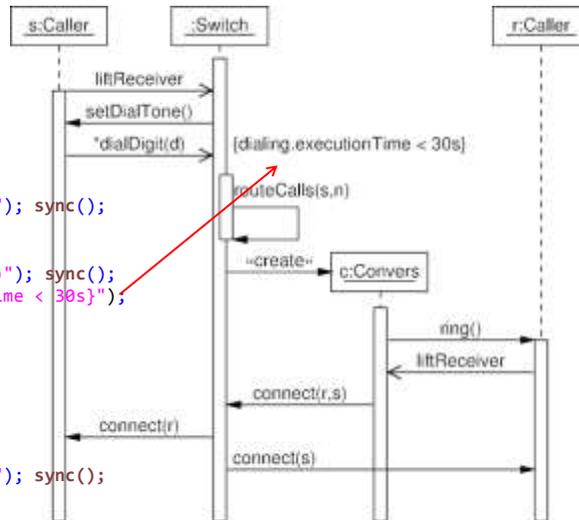




```

movewidth = 0.5;
# Define the objects
object(S, "s:Caller");
object(W, ":Switch");
object(C);
object(R, "r:Caller");
# Message sequences
step();
active(W);
async(); message(S,W,"liftReceiver"); sync();
active(S);
message(W,S,"setDialTone()");
async(); message(S,W,"*dialDigit(d)"); sync();
lconstraint(W, "{dialing.executionTime < 30s}");
active(W);
message(W,W,"routeCalls(s,n)");
inactive(W);
cmessage(W,C,"c:Convers");
active(C);
message(C,R,"ring()");
connect(r);
active(R);
async(); message(R,C,"liftReceiver"); sync();
message(C,W,"connect(r,s)");
message(W,S,"connect(r)");
# Specify label as a "constraint" to align on W
message(W,R,""); lconstraint(W,"connect(s)");
step();
complete(S);
complete(W);
complete(C);
complete(R);

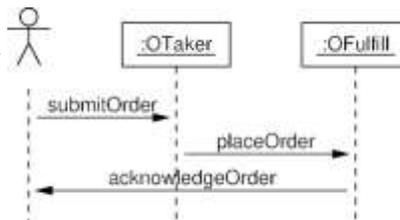
```

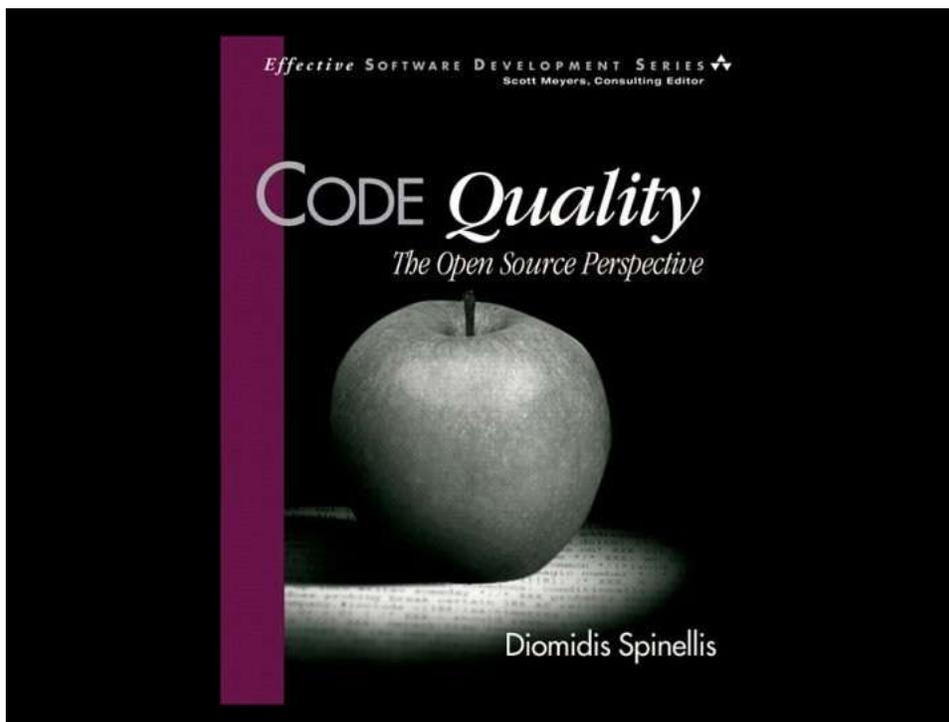
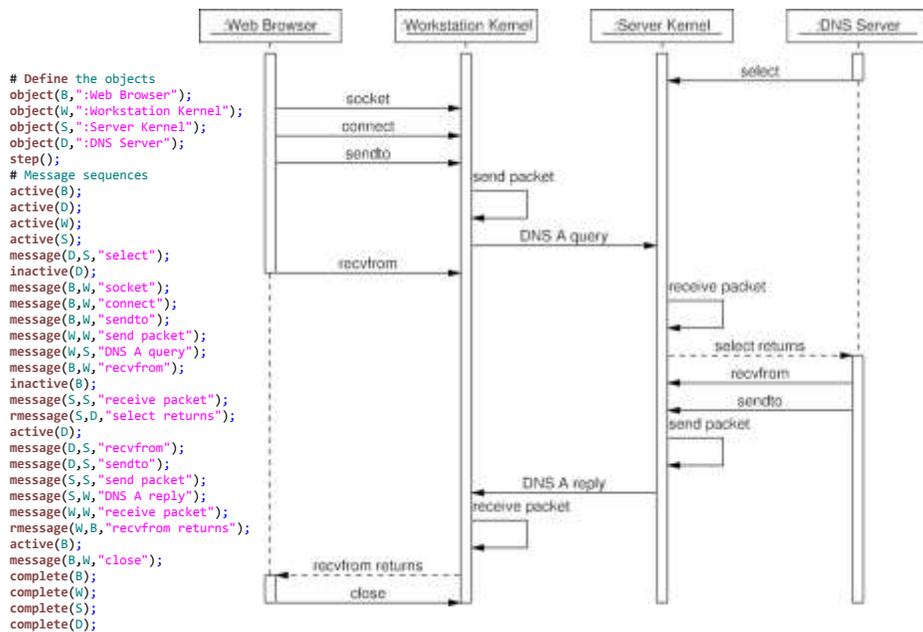


```

actor(A, "");
object(T, ":OTaker");
object(F, ":OFulfill");
step();
message(A,T,"submitOrder");
message(T,F,"placeOrder");
message(F,A,"acknowledgeOrder");
step();
complete(A);
complete(T);
complete(F);

```

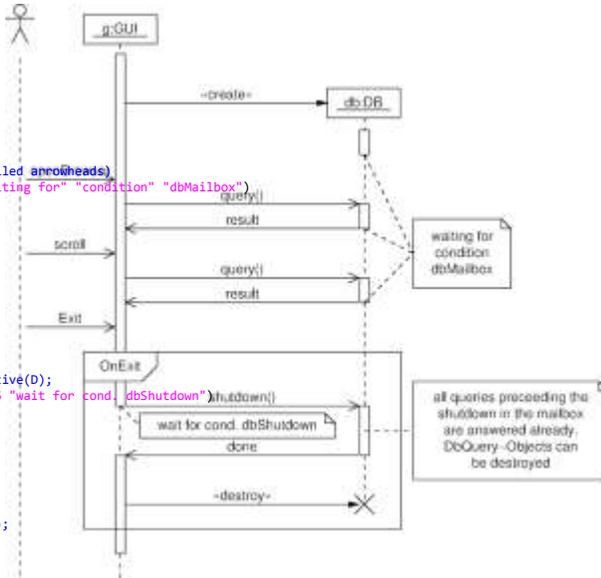




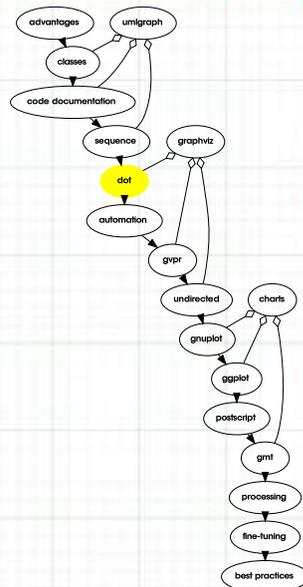
```

# rename the objects
actor(U, "");
object(G, "g:GUI");
placeholder_object(Dummy1); # more space
placeholder_object(D);
step();
# Message sequences
active(G);
step();
create_message(G,D,"db:DB");
active(D);
step();
inactive(D);
async(); # use asynchrone messages (not-filled arrowheads)
comment(D,C,down 1 right, wid 1 ht 0.7 "waiting for" "condition" "dbMailbox");
message(U,G,"openBrowser");
message(G,D,"query()"); active(D);
message(D,G,"result"); inactive(D);
connect_to_comment(D,C)
message(U,G,"scroll");
message(G,D,"query()"); active(D);
message(D,G,"result"); inactive(D);
connect_to_comment(D,C)
message(U,G,"Exit");
step();
begin_frame(G,F,"OnExit");
message(G,D,"shutdown()"); inactive(G); active(D);
comment(G,C,down .2 right .2, wid 2 ht 0.25 "wait for cond. dbShutdown");
step();
comment(D,C,right, wid 2 ht 1 \
  "all queries preceeding the" \
  "shutdown in the mailbox" \
  "are answered already." \
  "DbQuery-Objects can" \
  "be destroyed")
message(D,G,"done"); inactive(D); active(G);
sync();
destroy_message(G,D,"");
step();
end_frame(D,F);
step();
inactive(G);
# Complete the lifelines
step();
complete(G);
complete(D);

```



dot

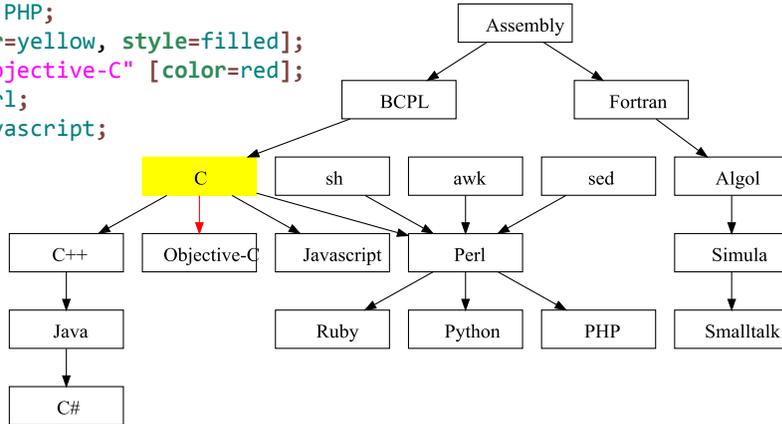




```

digraph {
  node [shape=box, width=1.5];
  Assembly -> BCPL -> C -> "C++" -> Java -> "C#";
  Assembly -> Fortran -> Algol -> Simula -> Smalltalk;
  sh -> Perl;
  awk -> Perl;
  sed -> Perl;
  Perl -> Ruby;
  Perl -> Python;
  Perl -> PHP;
  C [color=yellow, style=filled];
  C -> "Objective-C" [color=red];
  C -> Perl;
  C -> Javascript;
}

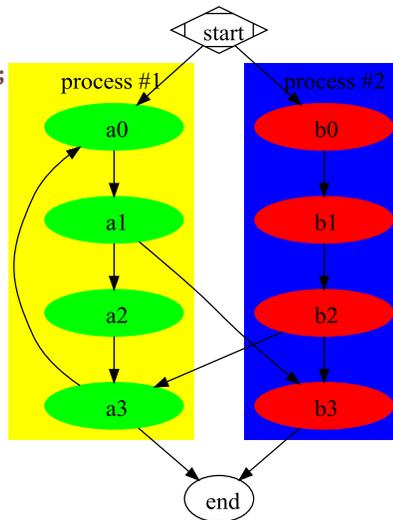
```

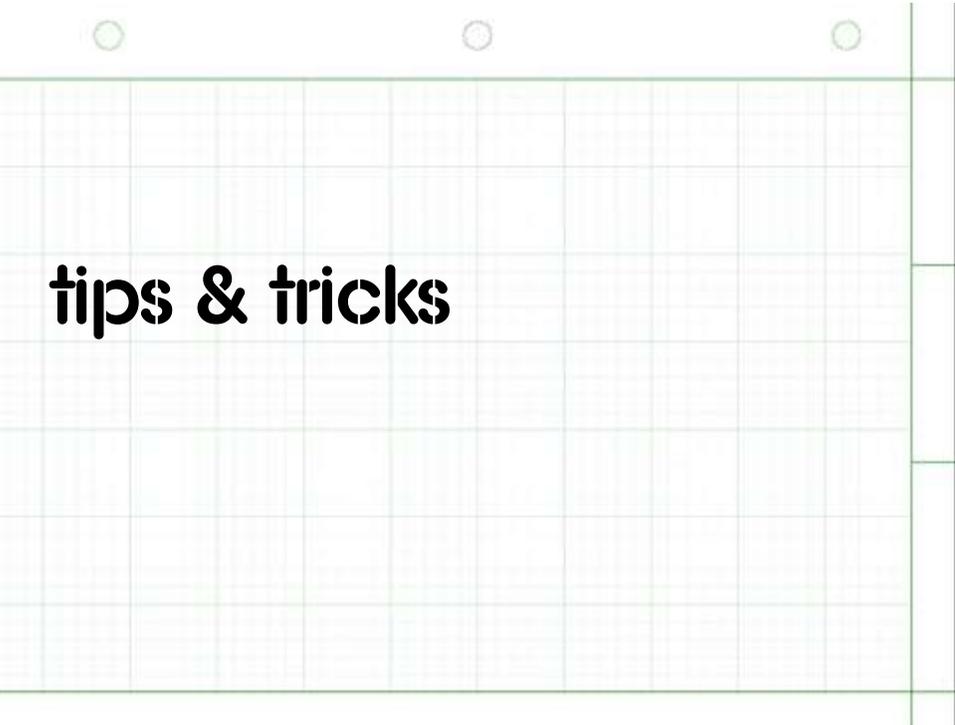


```

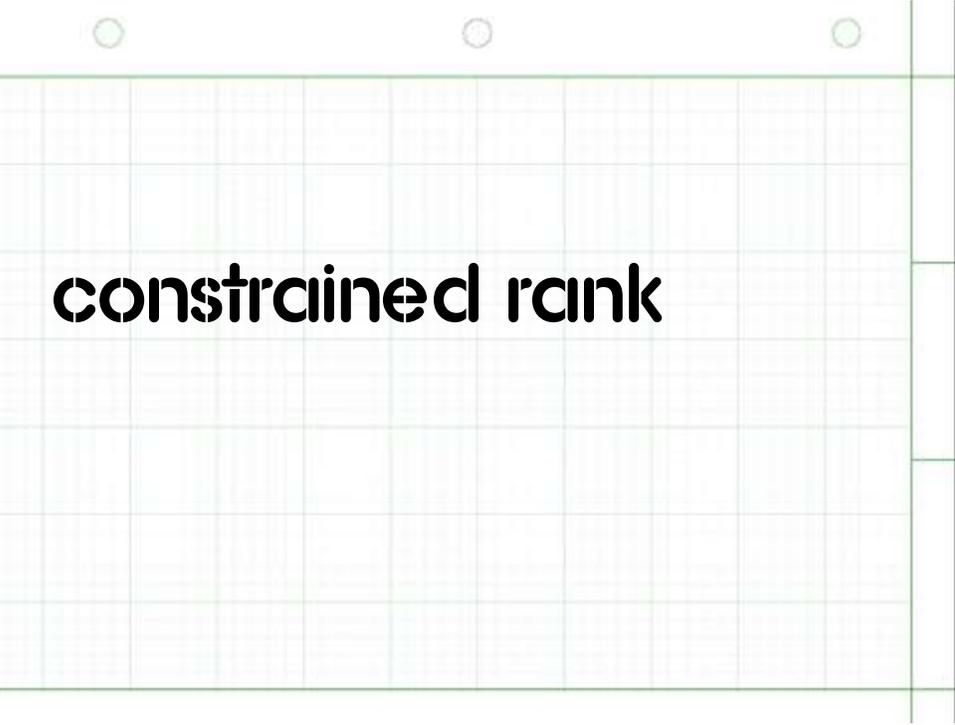
digraph G {
  subgraph cluster0 {
    node [style=filled,color=green, width=1.5];
    style=filled;
    color=yellow;
    a0 -> a1 -> a2 -> a3;
    label = "process #1";
  }
  subgraph cluster1 {
    node [style=filled, color=red, width=1.5];
    b0 -> b1 -> b2 -> b3;
    label = "process #2";
    style=filled;
    color=blue;
  }
  start -> a0;
  start -> b0;
  a1 -> b3;
  b2 -> a3;
  a3 -> end;
  b3 -> end;
  start [shape=Mdiamond];
  end
}

```



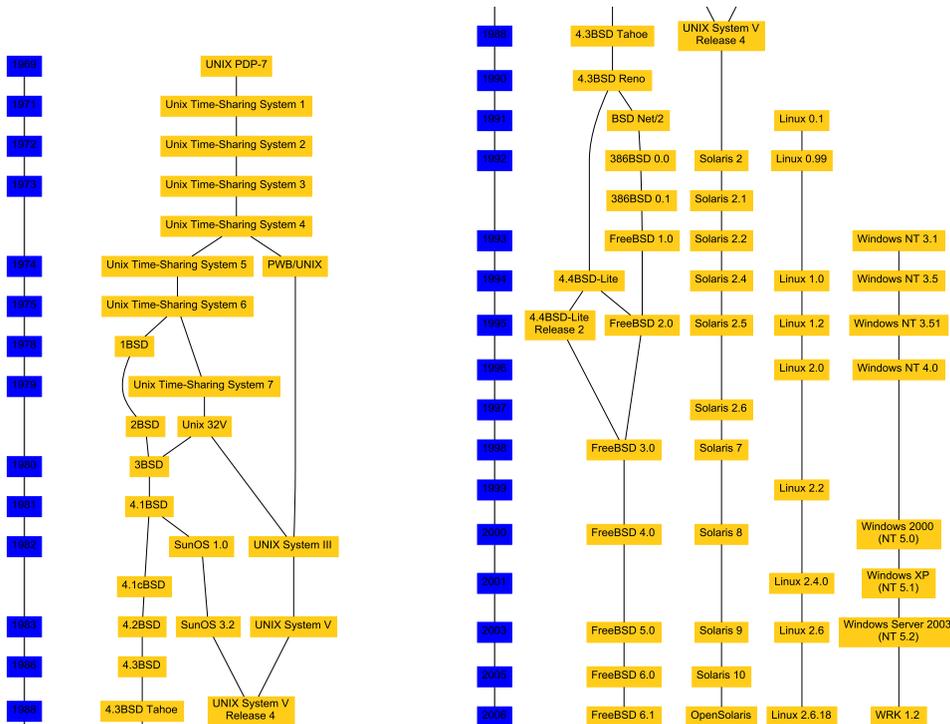


**tips & tricks**



**constrained rank**





```

digraph unix {
    node [fontsize=20, style="filled", color=".13 .9 1",
        fontname="c:/windows/fonts/comic.ttf"];
    size="8,14";
    {
    edge [style=bold, arrowhead=none];
    node [fillcolor=blue, style=filled, shape=plaintext,
        fontsize=20,fontname="Arial"];
    "1969" ->
    "1971" ->
    "1972" ->
    "1973" ->
    "1974" ->
    [...]
    "2001" ->
    "2003" ->
    "2005" ->
    "2006";
    }
}

```

```

{
edge [style=invis];
1973.00 [style=invis,height=0.1,width=0.0,fontsize=2];
1973.02 [style=invis,height=0.1,width=0.0,fontsize=2];
1974.00 [style=invis,height=0.1,width=0.0,fontsize=2];
1973.00 ->
1973.02 ->
1974.00;
}
{ rank=same; "1973"; "1973.00"; }
{ rank=same; "1974"; "1974.00"; }

```

```

{
node [shape=box, fontname="Arial", fontsize=20];
edge [style=dashed,arrowhead=none,arrowtail=open];
"UNIX PDP-7" -> "Unix Time-Sharing System 1" ->
"Unix Time-Sharing System 2" ->
"Unix Time-Sharing System 3" ->
"Unix Time-Sharing System 4" ->
"Unix Time-Sharing System 5" ->
"Unix Time-Sharing System 6";

"Unix Time-Sharing System 4" ->
"PWB/UNIX" ->
"UNIX System III" ->
"UNIX System V" ->
"UNIX System V\nRelease 4";

"Unix Time-Sharing System 6" -> "1BSD";

"Unix Time-Sharing System 6" ->
"Unix Time-Sharing System 7" ->
"Unix 32V" -> "UNIX System III";

"Unix 32V" -> "3BSD";
"2BSD" -> "3BSD";
"1BSD" -> "2BSD";

```

```

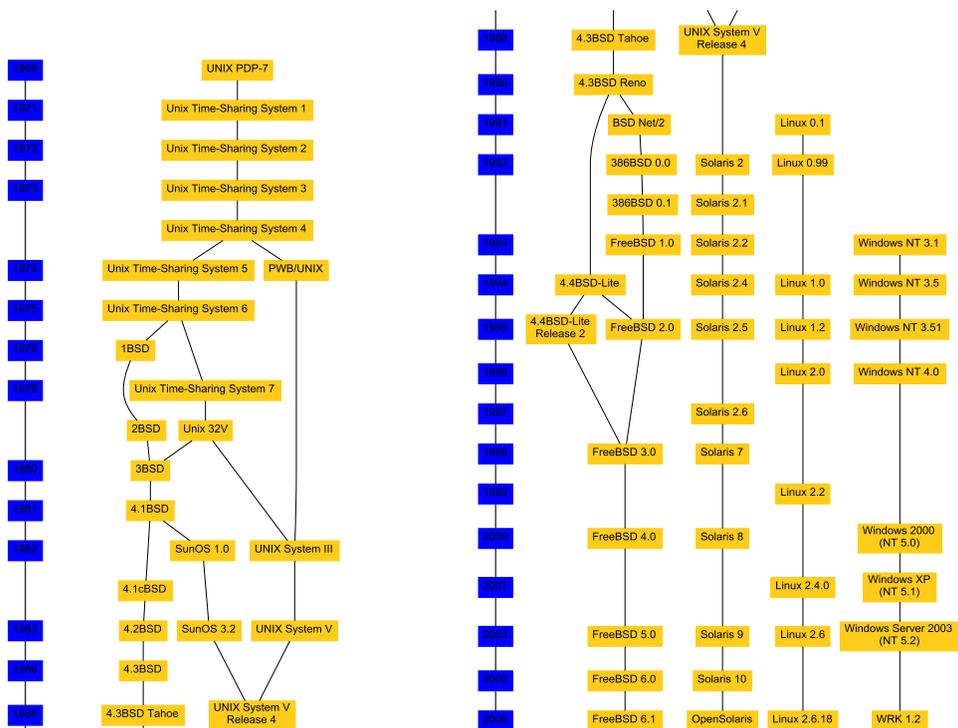
{rank=same; "1969"; "UNIX PDP-7";}
{rank=same; "1971"; "Unix Time-Sharing System 1";}
{rank=same; "1972"; "Unix Time-Sharing System 2";}
{rank=same; "1973"; "Unix Time-Sharing System 3";}
{rank=same; "1973.02"; "Unix Time-Sharing System 4";}

```

```

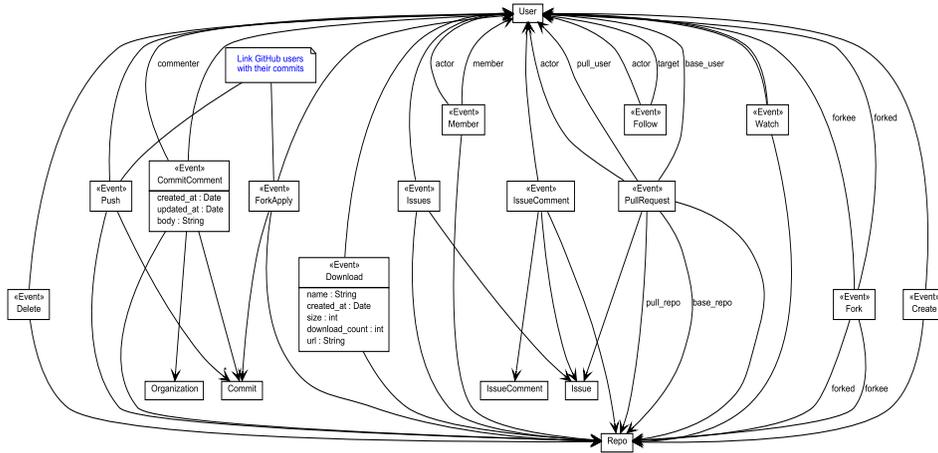
{rank=same; "2006"; "FreeBSD 6.1"; "Linux 2.6.18"; "WRK 1.2"; }

```





```
{ rank=source; User; }
{ rank=sink; Repo; }
```



postscript library

```

/densedashed {
    [2 InvScaleFactor mul dup ] 0 setdash
} bind def

% Decorate the shape with a UML package hat
/xdef {exch def} bind def
/UMLpackage {
    20 dict begin
        /fflag xdef
        /sides xdef
        dup
        aload pop
        pop pop
        /y1 xdef /x1 xdef
        /y2 xdef /x2 xdef
        /y3 xdef /x3 xdef
        /y4 xdef /x4 xdef
        newpath
        x3 y3 moveto
        x3 y3 5 add lineto
        x3 12 add y3 5 add lineto
        x2 12 add y3 lineto
        stroke
        pop
    end
} bind def

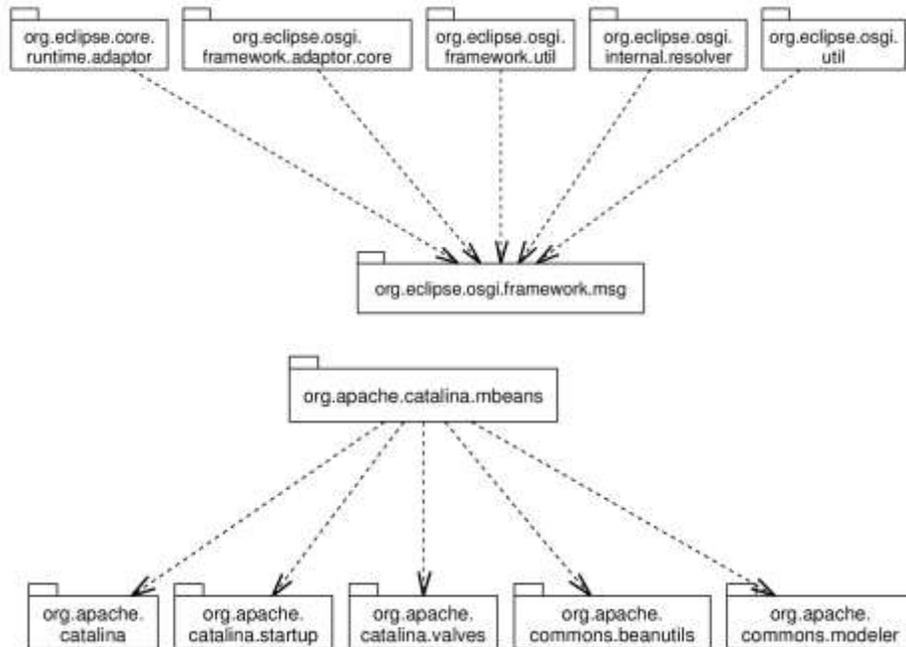
```

```

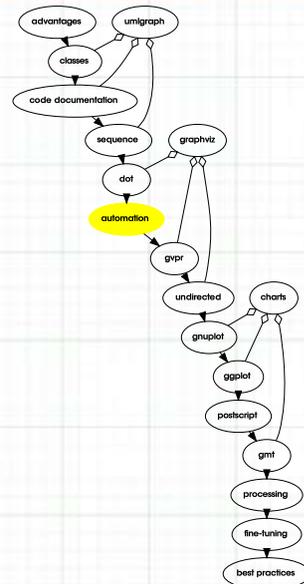
digraph G {
    node [shape=UMLpackage, height=0.3, fontname="Helvetica", fontsize=8];
    s [label="org.eclipse.osgi.framework.msg"];
    edge [arrowhead=open, arrowtail=none, style=densedashed];

    "org.eclipse.core.\nruntime.adaptor" -> s;
    "org.eclipse.osgi.\nframework.adaptor.core" -> s;
    "org.eclipse.osgi.\nframework.util" -> s;
    "org.eclipse.osgi.\ninternal.resolver" -> s;
    "org.eclipse.osgi.\nutil" -> s;
}

```



automation



```

digraph D { \
    ranksep=.1;
    node [fontname="AG Stencil", fontsize=10];
    // CURRENTNODE
    Advantages ->
    Classes ->
    "Code documentation" ->
    Sequence ->
    dot ->
    Automation ->
    gvpr ->
    Undirected ->
    gnuplot ->
    ggplot ->
    Postscript ->
    GMT ->
    Processing ->
    "Fine-tuning" ->
    "Best practices"; // For the script ->

    edge [arrowhead=none, arrowtail=odiamond, dir=both];
    UMLGraph -> Classes;
    UMLGraph -> Sequence;
    UMLGraph -> "Code documentation";
    GraphViz -> dot;
    GraphViz -> gvpr;
    GraphViz -> Undirected;
    Charts -> gnuplot;
    Charts -> ggplot;
    Charts -> GMT;
}

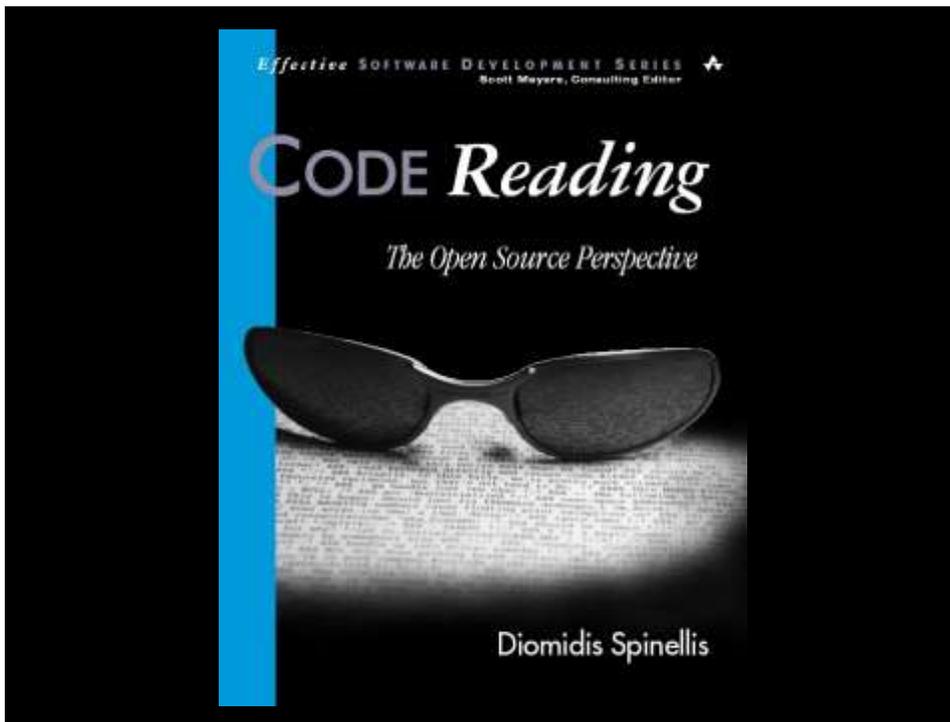
```

```

#!/bin/sh
# Create a series of highlighted scripts from outline.dot
sed -n '
    /->$/ {
        s/^[          ]*//
        s/ ->//
        s/;.*/
        s/"//g
        p
    }' outline.dot |
while read name
do
    (
        sed -n '1,/CURRENTNODE/p' outline.dot
        echo "\"$name\" ' [color=yellow, style=filled];'
        sed -n '/CURRENTNODE/, $p' outline.dot
    ) |
    dot -Temf -o"$name.emf"
done

```

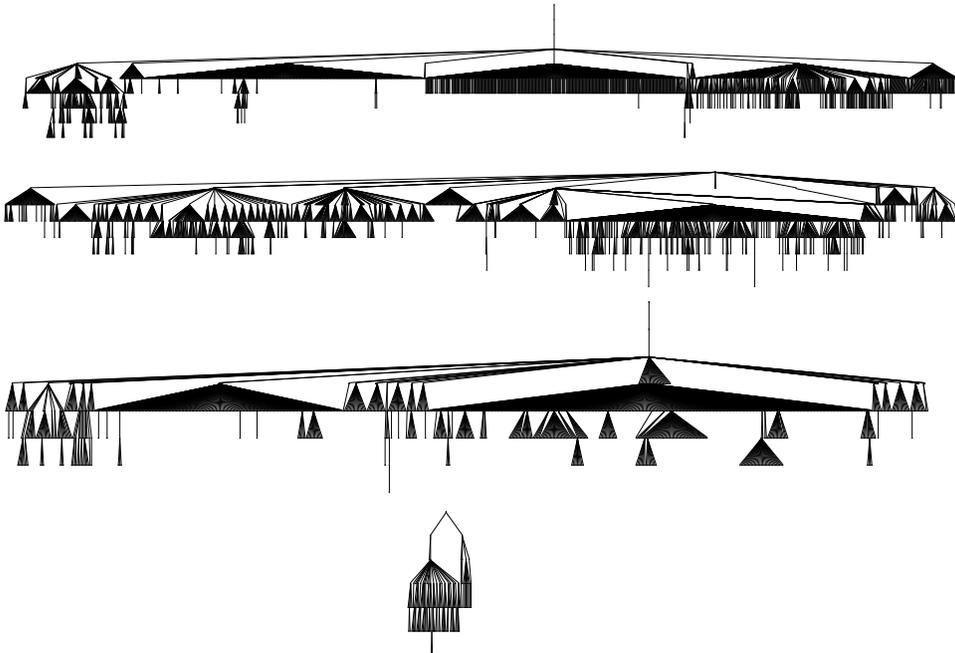
directories



```

open(IN, my $cmd = "find $ARGV[0] -type d -print|") ||
  die "Unable to run $cmd: $!\n";
while (<IN>) {
  chop;
  @path_elements = split(/\//, $_);
  my $parent;
  my $path;
  for my $element (@path_elements) {
    $path .= "/"$element;
    $name = $path;
    $name =~ s/[^a-zA-Z0-9]/_/g;
    $node{$name} = $element;
    $edge{"$parent->$name"} = 1 if ($parent);
    $parent = $name;
  }
}
close(IN);
print "digraph G {\n";
for my $i (sort keys %node) {
  print qq<\t$i [label="$node{$i}"];>\n";
}
for my $i (sort keys %edge) {
  print "\t$i\n";
}
print "}\n";

```



# phd genealogy

## Mathematics Genealogy Project

**Paul (Pál) Erdős**  
[Biography](#) [Math-Guide](#)

Ph.D. Eötvös Loránd University 1934 

Dissertation: Über die Primzahlen grosser arithmetischer Reihen

Advisor: [Leopold Lindöf Engel](#)

Students:  
Click [here](#) to see the students listed in chronological order

Name	School	Year	Descendants
<a href="#">Frédéric Bourqui</a>			11
<a href="#">Béla Bollobás</a>	Eötvös Loránd University	1967	113
<a href="#">Borisac Donat</a>	Ecole Normale Supérieure Polytechnique Antananariva	1967	
<a href="#">Borogai Karmakar</a>	Hungarian Academy of Sciences	1990	
<a href="#">Joseph Kruskal</a>	Princeton University	1954	
<a href="#">Alexander Soffer</a>	Moscow State Pedagogical Institute	1973	

According to our current on-line database, Paul Erdős has 6 [students](#) and 130 [descendants](#).  
We welcome any additional information.

If you have additional information or corrections regarding this mathematician, please use the [update form](#). To submit students of this mathematician, please use the [add data form](#).

[Search](#) [Extrema](#) [About MGP](#) [Links](#) [FAQs](#) [Posters](#) [Submit Data](#)

The Mathematics Genealogy Project is in need of funds to help pay for student help and other associated costs. If you would like to contribute, please [donate online](#) using credit card or bank transfer or mail your tax-deductible contribution to:

Mathematics Genealogy Project  
Department of Mathematics  
North Dakota State University  
P. O. Box 5830  
Fargo, North Dakota 58105-0830

```

use strict;
use LWP::Simple;

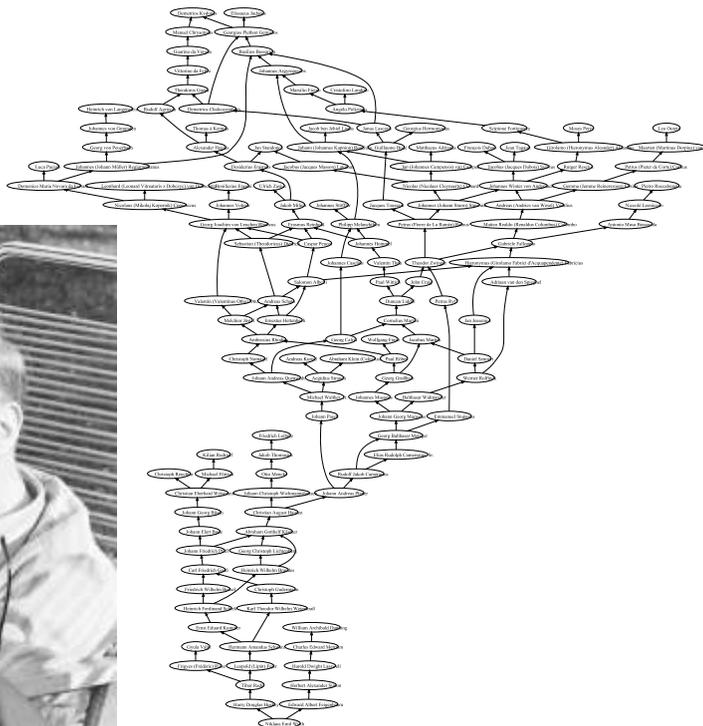
my %visited;
print 'digraph Genealogy {
    edge [dir = back];
';
trace(@ARGV);
print "}\n";

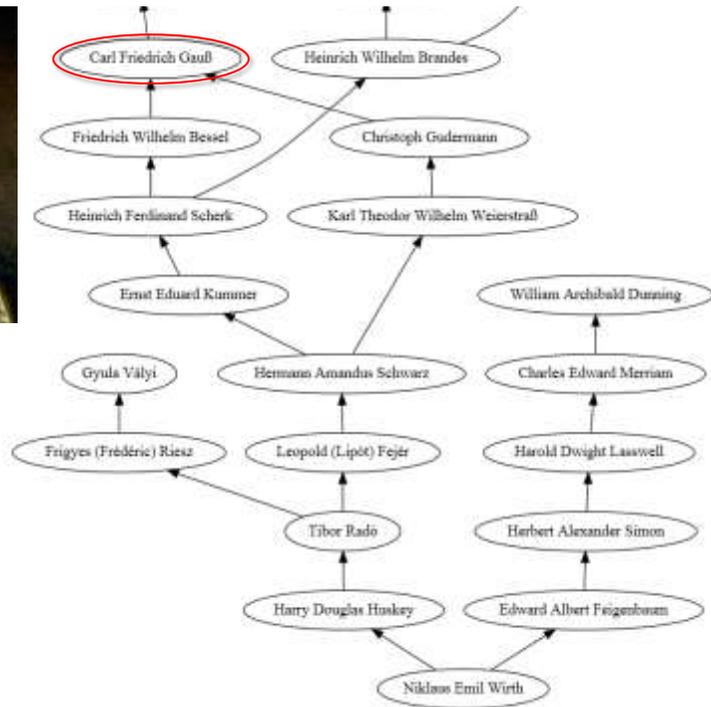
sub trace {
    my($id, $name) = @_;

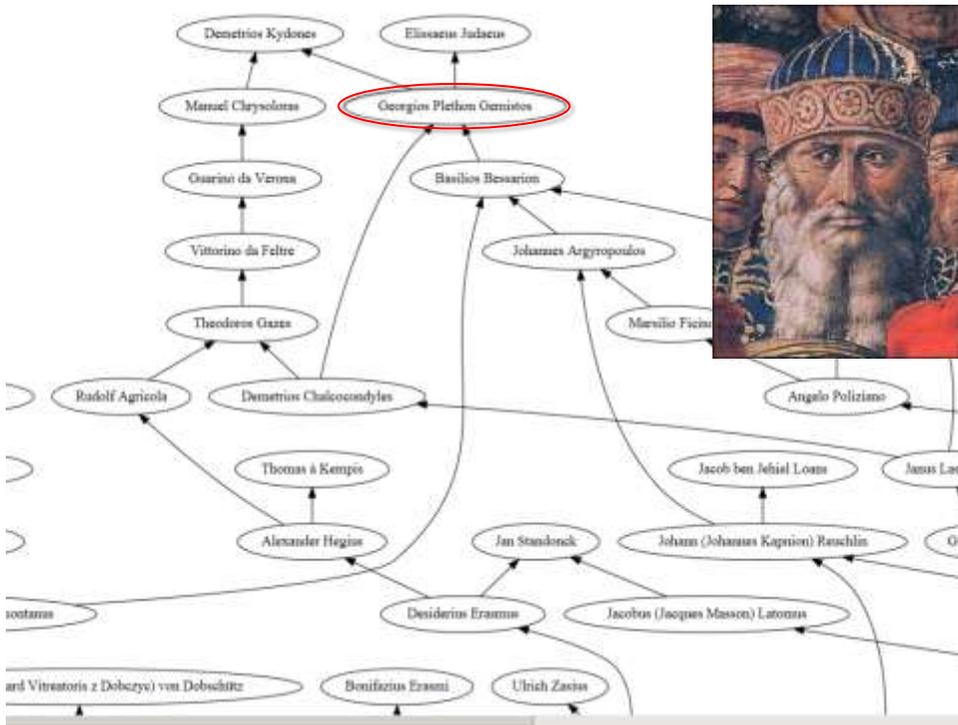
    return if ($visited{$id});
    print qq{\tn$id [label="$name"];\n};
    my $c = get("http://www.genealogy.math.ndsu.nodak.edu/id.php?id=$id");

    # Advisor 1: <a href="id.php?id=128059">Susan Eisenbach</a>
    while ($c =~ s/Advisor[^:]*: <a href="id\.php?id=(\d+)">([^\<]+)</a>/) {
        my ($aid, $aname) = ($1, $2);
        print qq{\tn$aid -> n$id;\n};
        trace($aid, $aname);
    }
    $visited{$id} = 1;
}

```





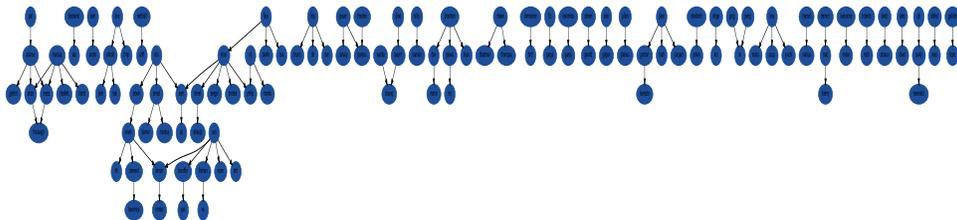


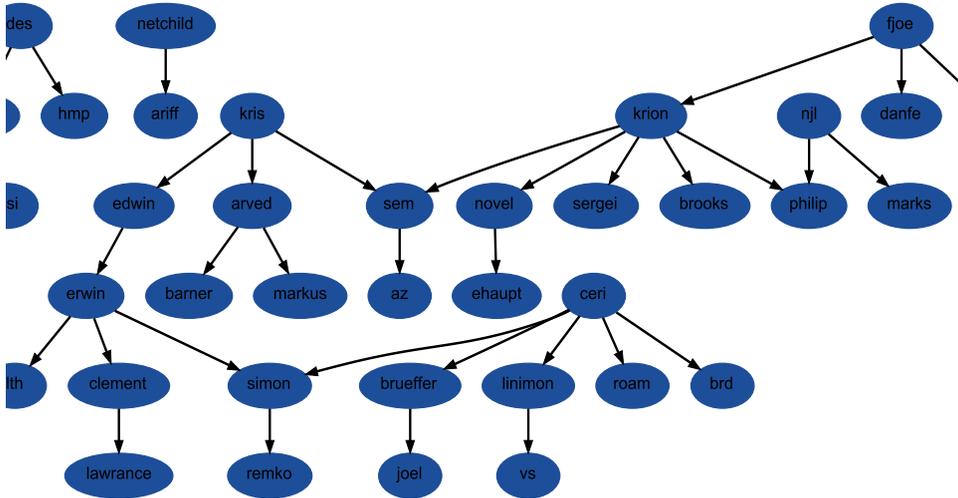
freebsd mentors

```

find src -type f |
grep -v CVS |
xargs cvs -d /home/ncvs log -SN 2>/dev/null
awk '
  /^date/ { author = $5 }
  /^Approved by:.*mentor/ { print author, $0 }' |
sed -n '
  s/co.mentor/mentor/
  s/backup //
  s/^\([^;]*\);.*^[a-z]\([a-z][a-z]*\) *(mentor.*\/\1 \2/p' |
sort -u |
awk '
  BEGIN { print "digraph mentors {" }
  { print $1 " -> " $2 ";" }
  END { print "}" }'

```





state transition

```

day_arm:
| call arm_init
| set_sensor_active("Door2", DELAYED)
| touch(STATUSPATH "dayarm")
> day_armed
;
day_armed:
| syslog(LOG_INFO, "Day armed")
DelayedSensor > check_entry
ActiveSensor > day_alarm
;
day_alarm:
| syslog(LOG_INFO, "Day alarm")
| increment_sensors()
| set("MainSiren", MAIN_ON)
| set("SmallSiren", SMALL_ON)
3s > day_rearm
;

```

```

static void
proc_ST_day_armed(void)
{
    state_count[ST_day_armed]++;
    syslog(LOG_DEBUG, "state: ST_day_armed (%d)", state_count[ST_day_armed]);
    syslog(LOG_INFO, "Day armed");

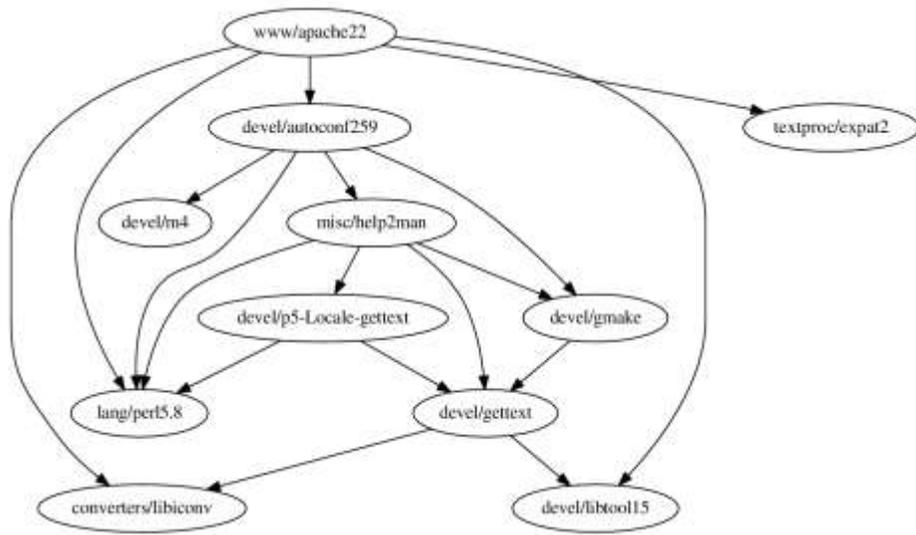
    for (;;) switch(get_event()) {
    case EV_CmdLeave: state = ST_leave; return;
    case EV_CmdDayArm: state = ST_day_arm; return;
    case EV_CmdNightArm: state = ST_night_arm; return;
    case EV_CmdVisitorArm: state = ST_visitor_arm; return;
    case EV_CmdDisarm: state = ST_disarm; return;
    case EV_CmdLoudPanic: state = ST_loud_panic; return;
    case EV_CmdSilentPanic: state = ST_silent_panic; return;
    case EV_CmdQuit: state = ST_exit; return;
    case EV_CmdTestSmall: state = ST_test_small; return;
    case EV_CmdTestMain: state = ST_test_main; return;
    case EV_DelayedSensor: state = ST_check_entry; return;
    case EV_ActiveSensor: state = ST_day_alarm; return;
    }
}

```



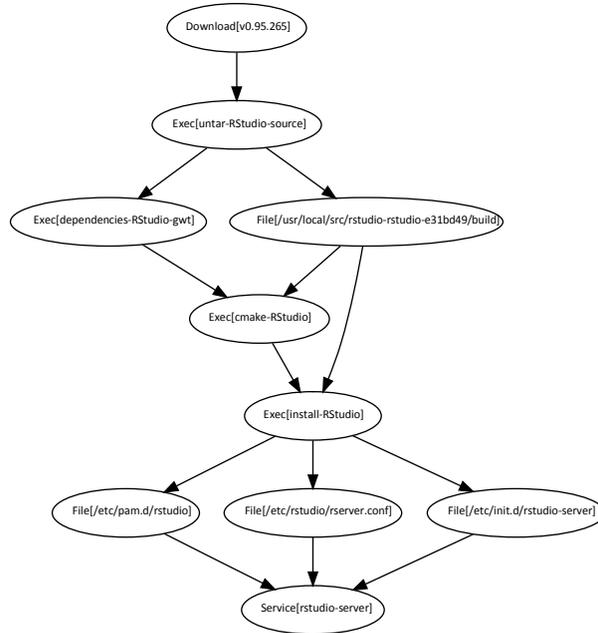
```
# New ports collection makefile for:    apache22
# Date created:                        Dec 11 2004
# Whom:                                Clement Laforet <clement@FreeBSD.org>
#
# $FreeBSD: ports/www/apache22/Makefile,v 1.285 2010/12/07 20:38:17 pgollucci
Exp $
#
PORTNAME=        apache
PORTVERSION=     2.2.17
MASTER_SITES=   ${MASTER_SITE_APACHE_HTTPD}
DISTNAME=       httpd-${PORTVERSION}
DIST_SUBDIR=    apache22
MAINTAINER?=    apache@FreeBSD.org
COMMENT?=       Version 2.2.x of Apache web server with ${WITH_MPM:L} MPM.
LIB_DEPENDS=    expat.6:${PORTSDIR}/textproc/expat2 \
                apr-1:${PORTSDIR}/devel/apr1 \
                pcre.0:${PORTSDIR}/devel/pcre
```

```
find /usr/ports -name Makefile -maxdepth 3 |
sed 's|/Makefile||' |
while read dir
do
    cd $dir
    make -V PORTNAME -V COMMENT -V RUN_DEPENDS \
        -V BUILD_DEPENDS -V LIB_DEPENDS \
        -V FETCH_DEPENDS -V DEPENDS && echo $dir
done
```

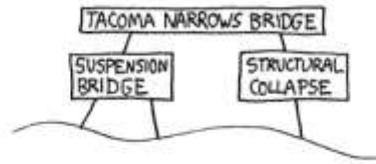


**puppet**

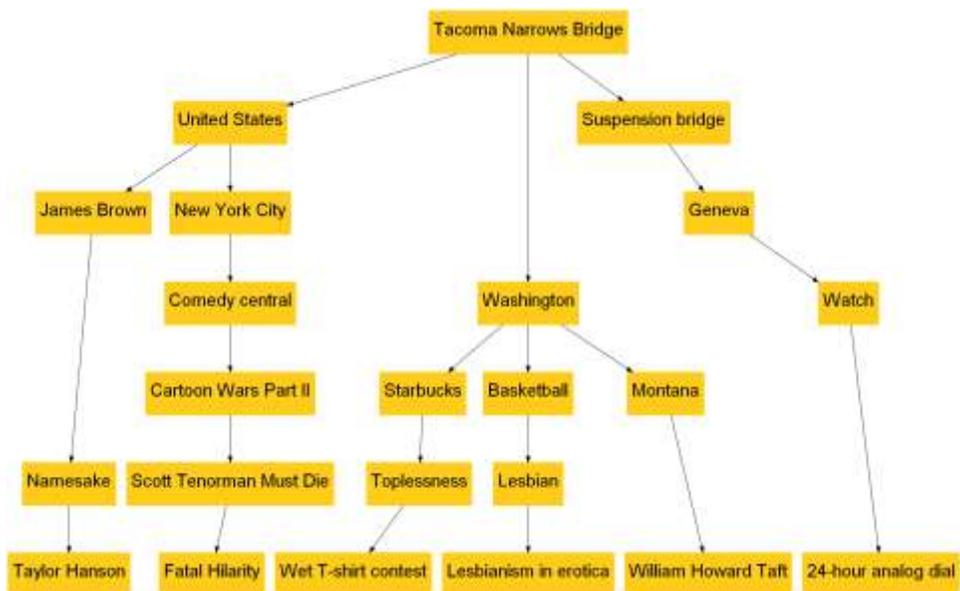
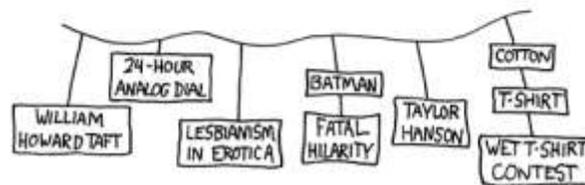
puppet apply --graph --graphdir /tmp site.pp

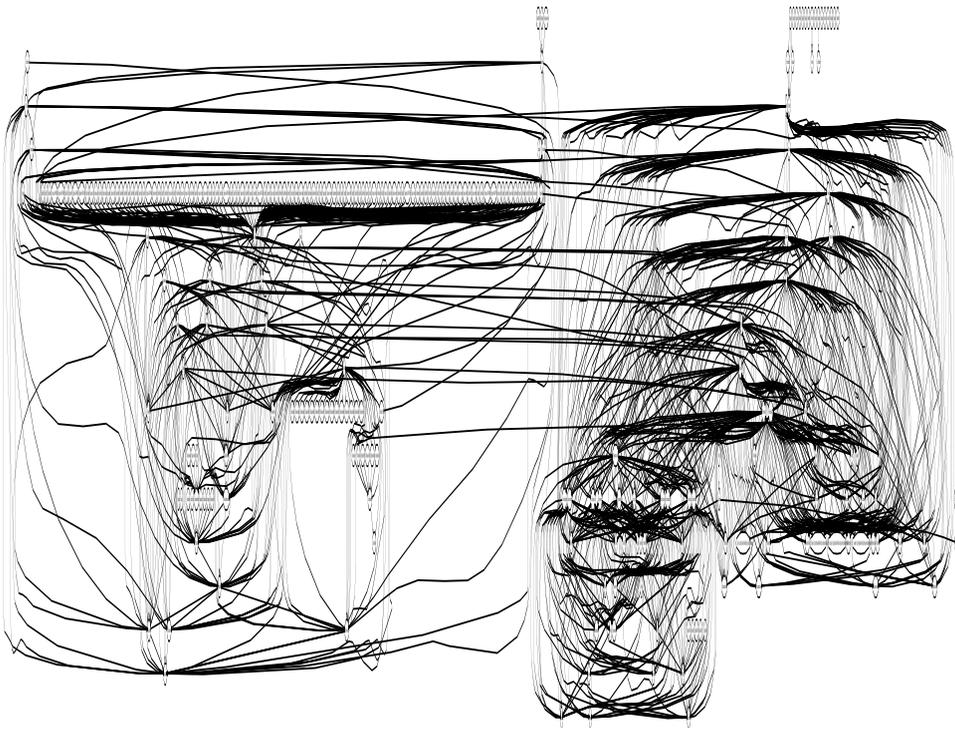


THE PROBLEM WITH WIKIPEDIA:

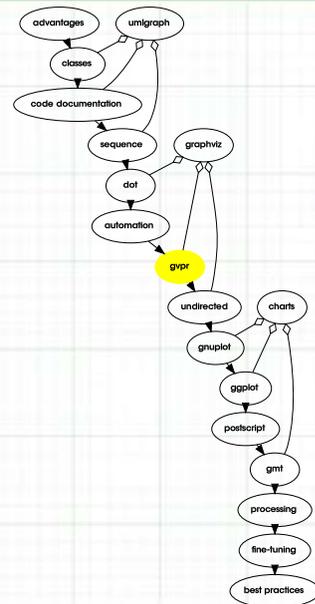


[THREE HOURS OF FASCINATED CLICKING]





**gvpr**



```

N {
    // Delete package and stand-alone nodes
    if (match(name, "[Pp]ackage") != -1) ||
        degreeOf($G, $) == 0)
        delete($G, $)
}
E {
    // Delete package edges
    if (match($.head.name, "[Pp]ackage") != -1 ||
        match($.tail.name, "[Pp]ackage") != -1)
        delete($G, $)
}
END_G { $O= $G }

```

```

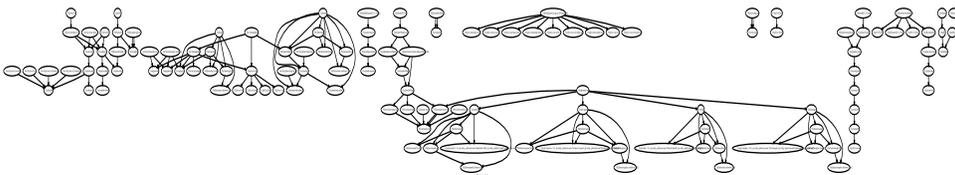
gvpr -f pack.gvpr relationships.dot |
dot -Tmf -o relationships.emf

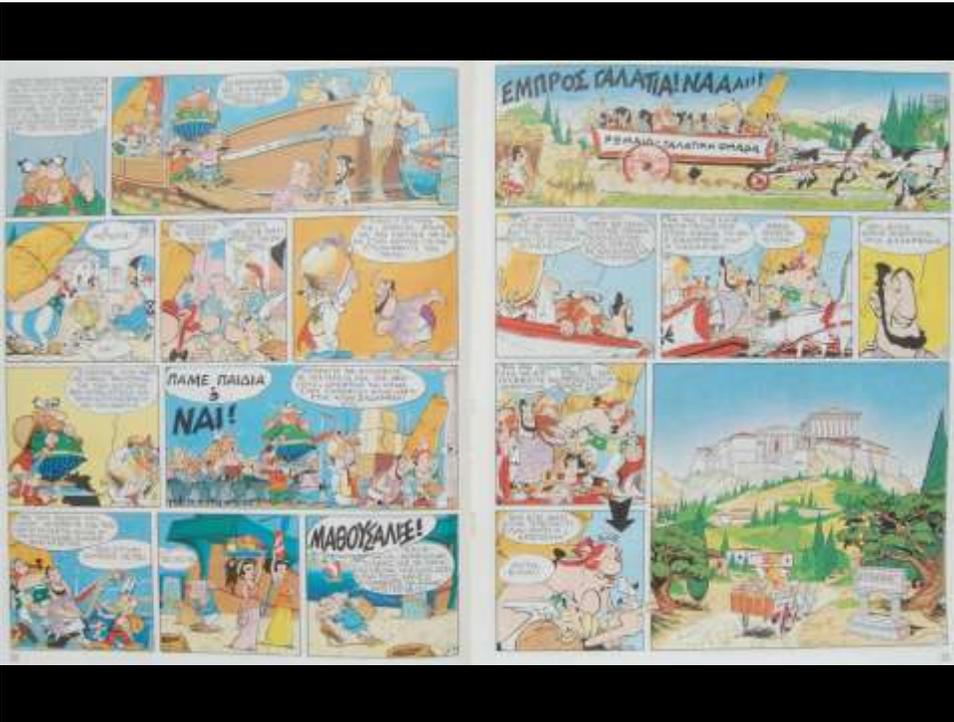
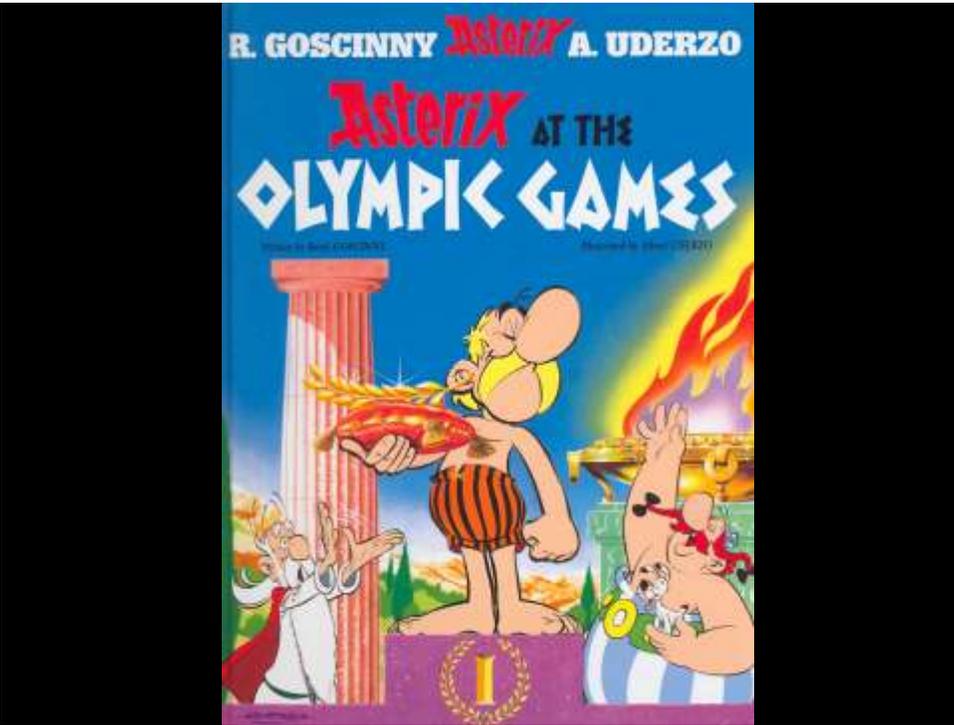
```

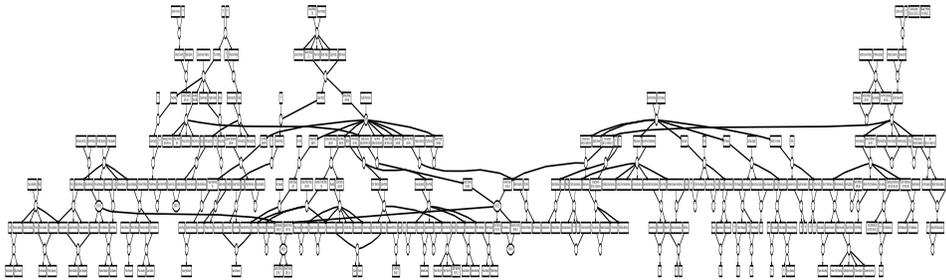
Before



After





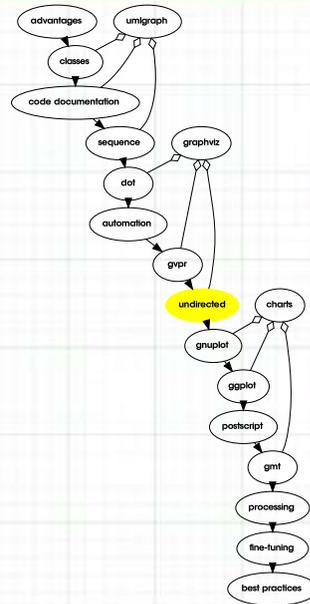


```

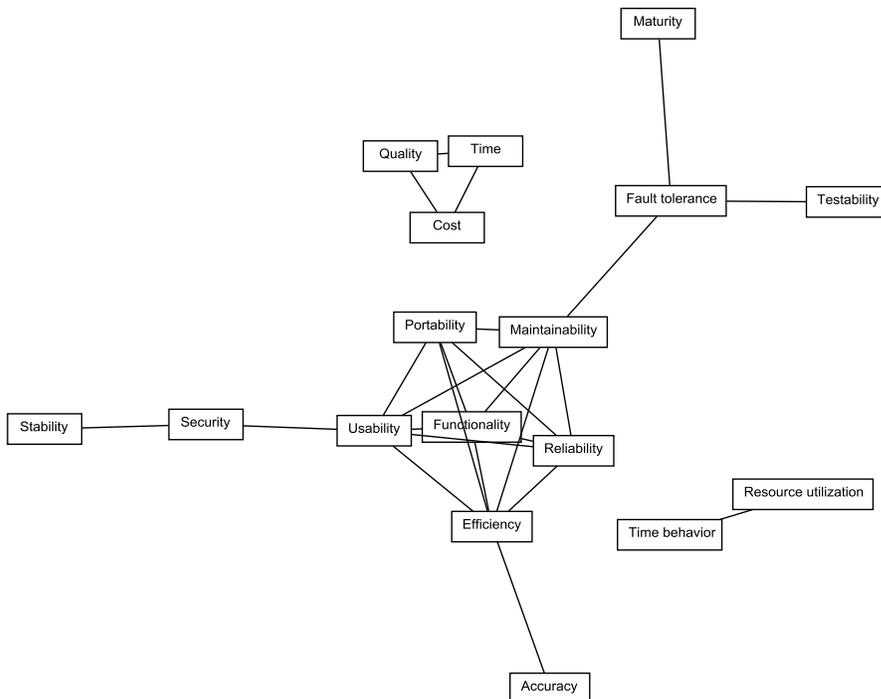
BEGIN {
    void usage(void) {
        printf(2, "Usage: subgraph start fwd|rev\n");
        exit(1);
    }
}
BEG_G {
    $tvroot = isNode($G, ARGV[0]);
    graph_t g = graph("merge", "S");
    if (ARGV[1] == "fwd")
        $tvtype = TV_fwd;
    else if (ARGV[1] == "rev")
        $tvtype = TV_rev;
    else
        usage();
}
E {
    $tvroot = NULL;
    clone(g, $);
}
END_G { $0 = g; }

```

# undirected



```
graph G {
  start=6;
  overlap=false;
  node [shape=box,height=0.3,fontname="Helvetica",fontsize=10];
  Functionality -- Reliability;
  Functionality -- Usability;
  Functionality -- Efficiency;
  Functionality -- Maintainability;
  Functionality -- Portability;
  Reliability -- Usability;
  Reliability -- Efficiency;
  Reliability -- Maintainability;
  Reliability -- Portability;
  Usability -- Efficiency;
  Usability -- Maintainability;
  Usability -- Portability;
  Efficiency -- Maintainability;
  Efficiency -- Portability;
  Maintainability -- Portability;
  // Minor
  Accuracy -- Efficiency;
  Security -- Usability;
  Security -- Stability;
  Maturity -- "Fault tolerance";
  Maintainability -- "Fault tolerance";
  Testability -- "Fault tolerance";
  "Time behavior" -- "Resource utilization";
  Quality -- Time -- Cost -- Quality;
}
```



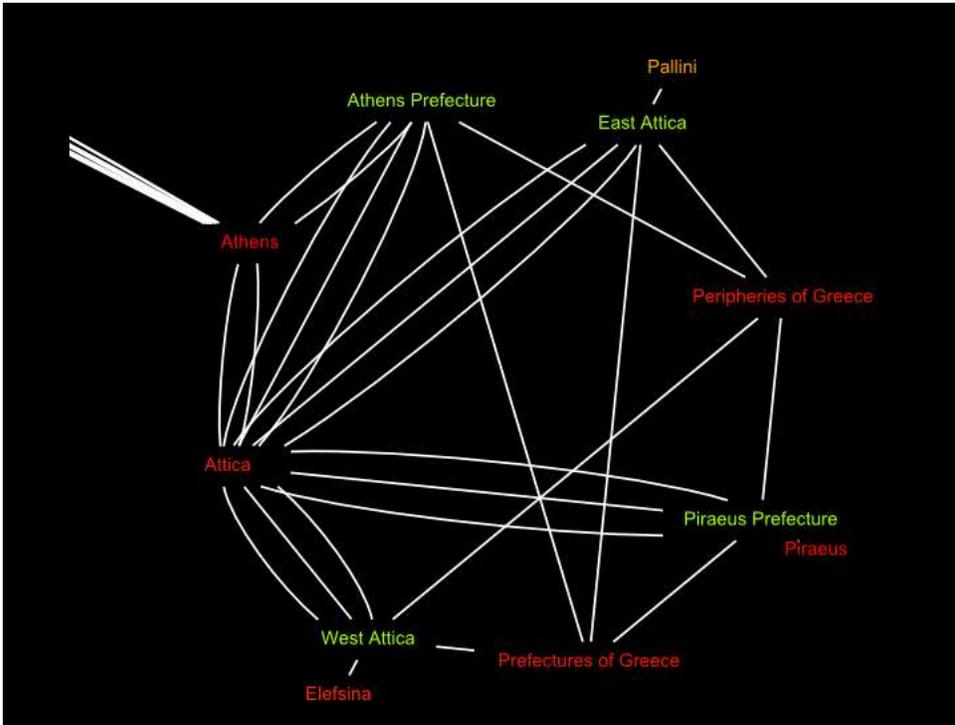
```

use LWP;
my $browser = LWP::UserAgent->new;
print "graph Wikipedia {\n";
crawl($ARGV[0], 0);
print "}\n";
sub crawl {
    my($name, $depth) = @_;
    return if ($done{$name});
    $done{$name} = 1;
    my $url = "http://en.wikipedia.org/wiki/Special:Export/$name";
    my $response = $browser->get($url);
    my $content = $response->content;
    print qq{\t"$name";\n};
    return if ($depth++ == $maxdepth);
    my $count = 0;
    while ($content =~ s/\[([.*?)(\]\)]|\[\..*?\]\]/DONE/) {
        my $newname = $1;
        next if ($newname =~ /[#:]/);
        last if ($count++ == $limit);
        # Change first character to uppercase
        $newname =~ s/^(.)/uc($1)/e;
        print qq{\t"$name" -- "$newname";\n};
        crawl($newname, $depth);
    }
}

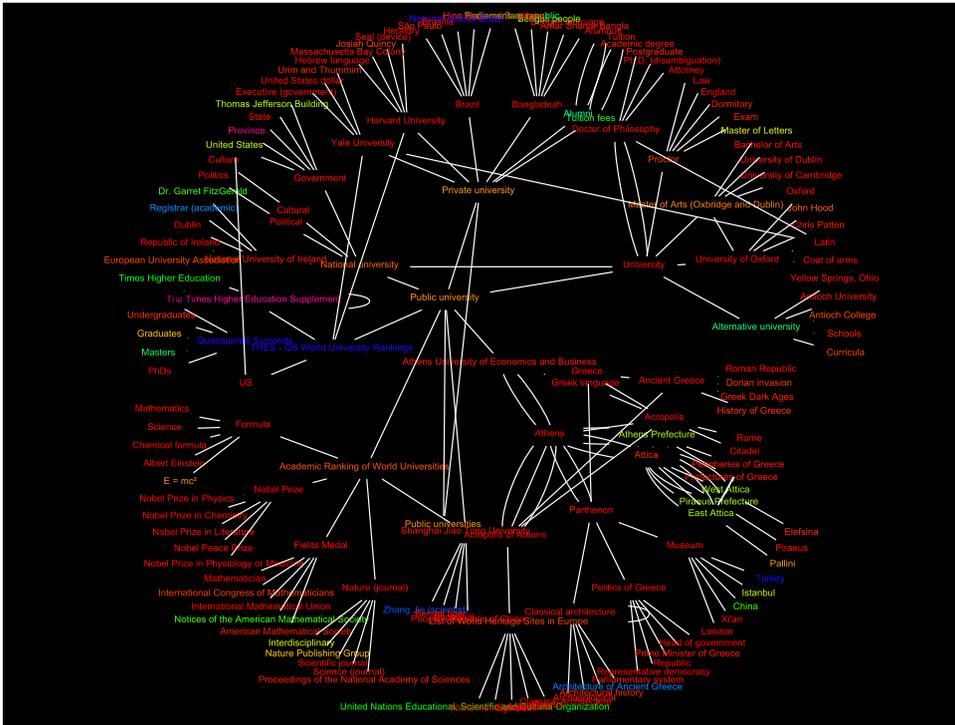
```



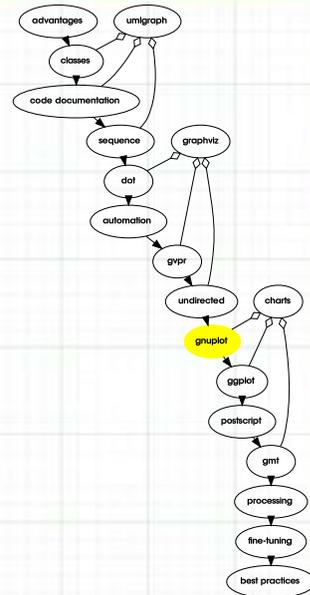


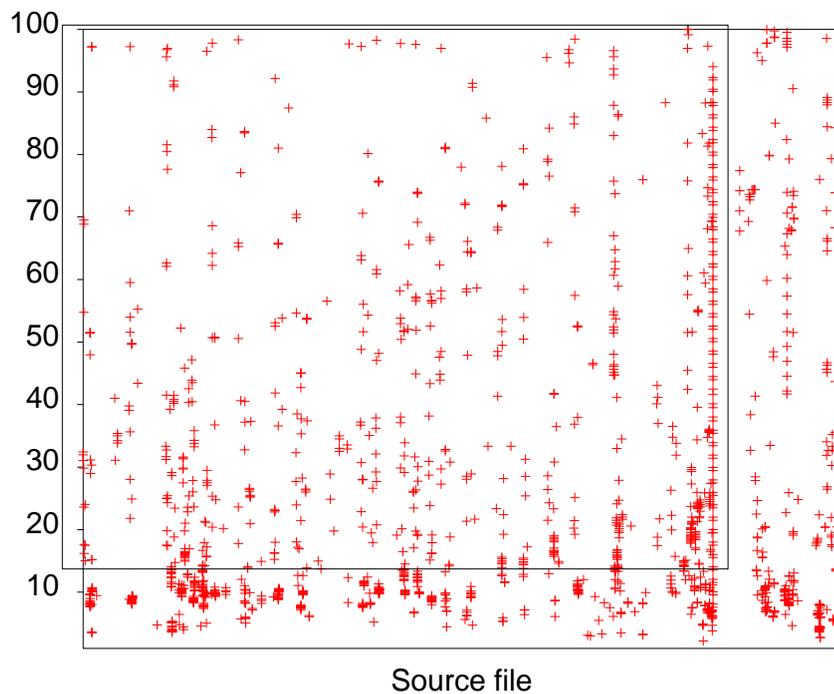


twopi



gnuplot





```
nm -l kernel |
grep ' [BDGRS] ' |
awk '{print $3}' |
sed 's/_//>' >kernsyms.dat
```

```
find /usr/src/sys -type f >sys.lst
```

```
perl kerndefpc.pl kernsyms.dat sys.lst >kerndefpc.gpl
```

```
perl kerndefpc.pl \
<(nm -l kernel | grep ' [BDGRS] ' | awk '{print $3}' | sed 's/_//') \
<(find /usr/src/sys -type f) >kerndefpc.gpl
```

```

# Compile pattern
my $patf = shift(@ARGV);
open(IN, $patf) || die;
while (<IN>) {
    chop;
    $pat .= "$_ |";
}
chop $pat;

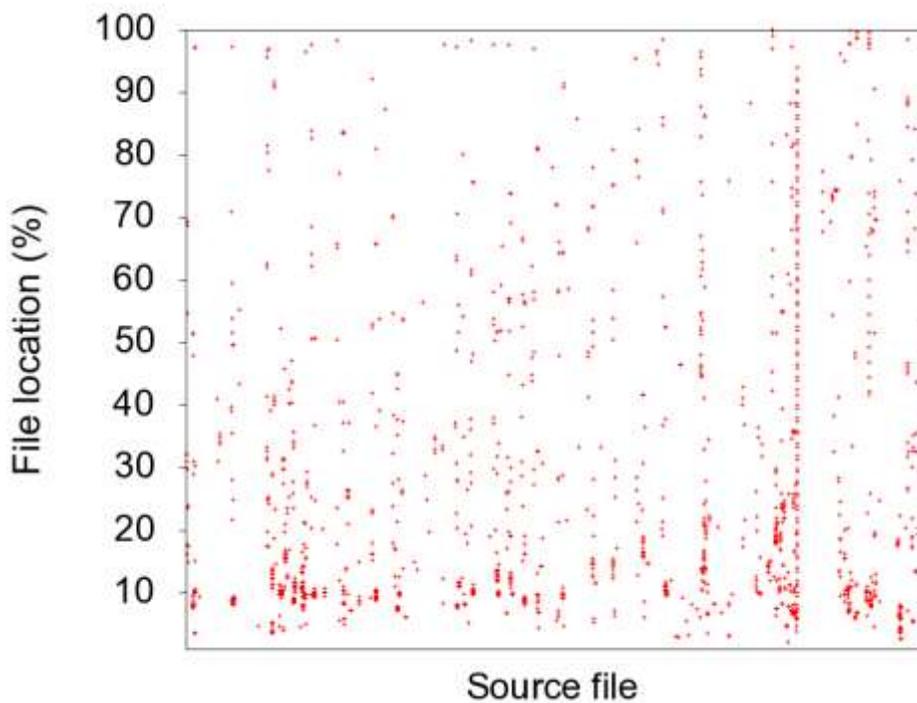
```

```

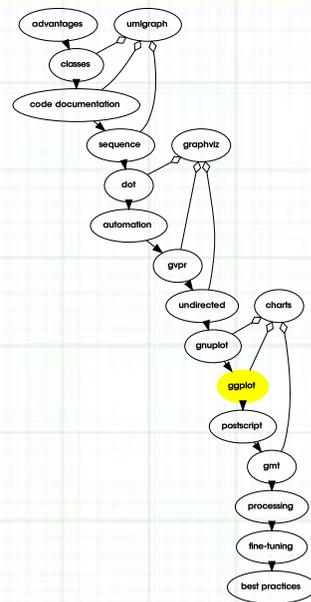
my $i;
# Read file names
while (<>) {
    chop;
    next if (/\.h$/);
    open(F, my $fname = $_) || die "Unable to open $_: $!";
    $i++;
    # Read the complete file
    my @lines = <F>;
    $nlines = $#lines + 1;
    my $line = 0;
    foreach $_ (@lines) {
        $line++;
        # Skip declarations
        next if (/extern/);
        next if (/^struct \w+ \{$/);
        # Match pattern as global definition
        if (($_ =~ m/^[^\s#\*\|/].*\b($pat)\b/o)) {
            print "$i ", $line / $nlines * 100, "\n";
        }
    }
    close(F);
}

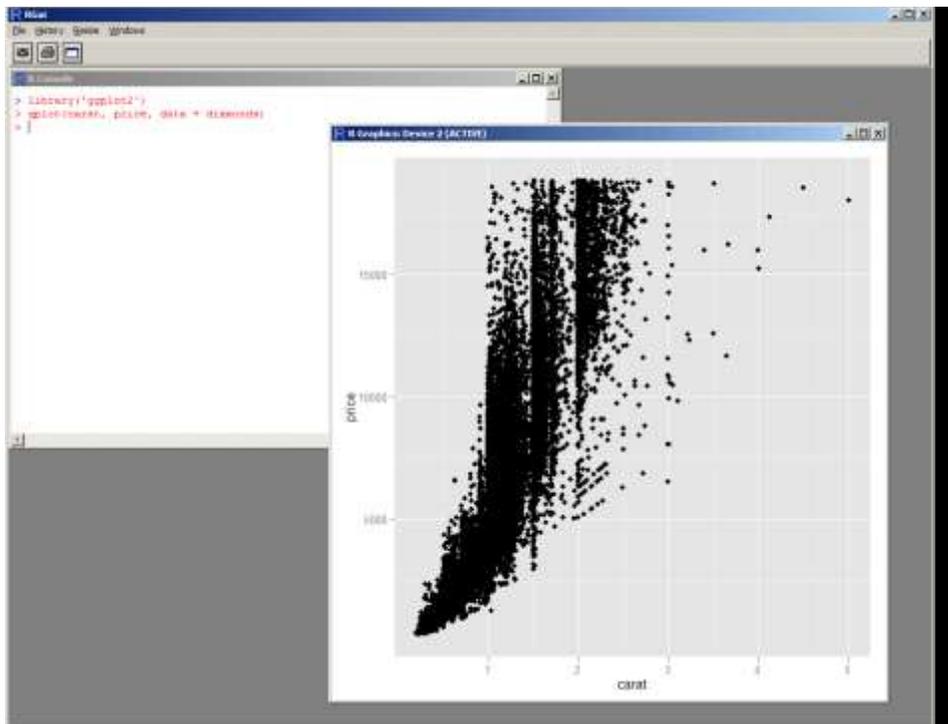
```

```
set xlabel 'Source file'
set ylabel 'File location (%)'
set tics out
set noxtics
set terminal png size 1024, 768 font 'Arial' 30
set pointsize .8
set output 'kerndefpc.png'
plot [] [1:100] '-' notitle with points
5 31.0502283105023
5 31.9634703196347
5 32.4200913242009
7 23.6065573770492
7 29.8360655737705
[...]
2182 82.3394495412844
2186 36.2043795620438
2186 38.6861313868613
2186 86.7153284671533
e
set output 'nul'
```



# R / ggplot2



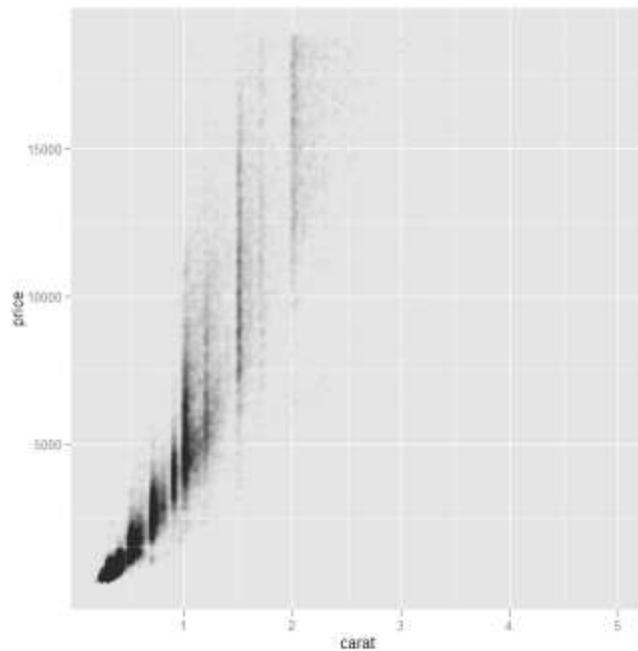


```
install.packages(c("ggplot2", "plyr"))
```

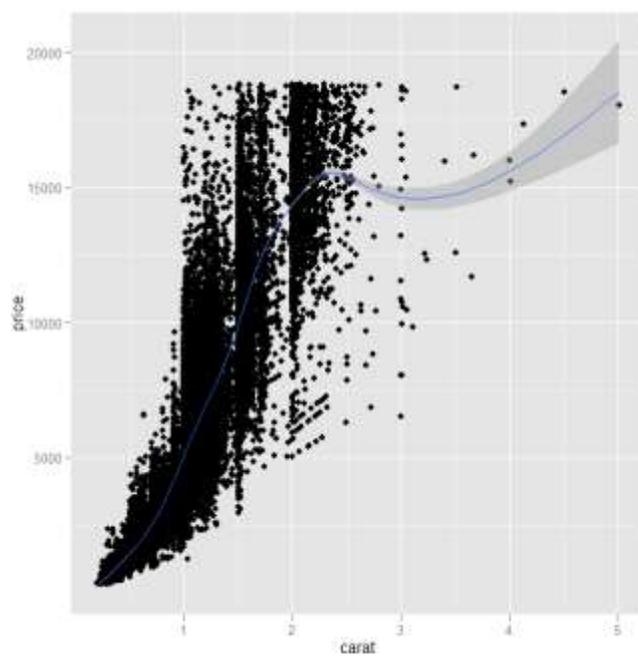
```
library('ggplot2')
qplot(carat, price, data = diamonds, alpha = I(1/100))
ggsave('carat.svg')
```

```
R CMD BATCH --slave drawcmds.R
```

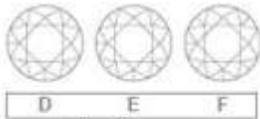
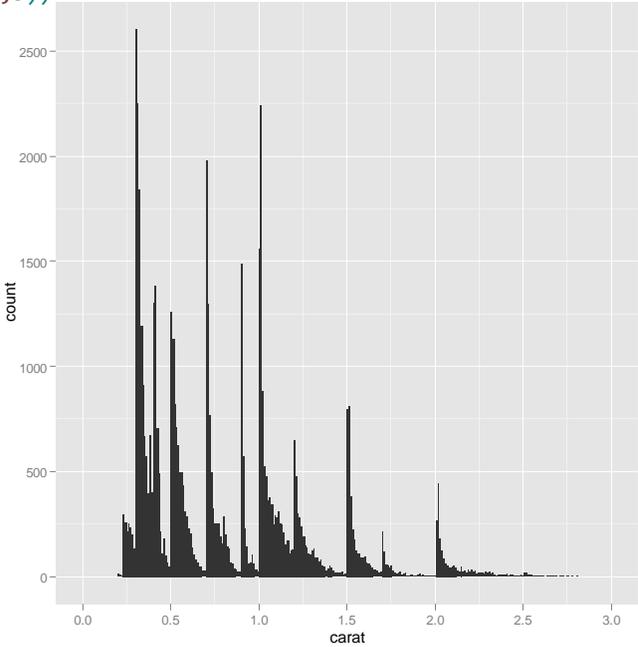
```
qplot(carat, price, data = diamonds, alpha = I(1/100))
```



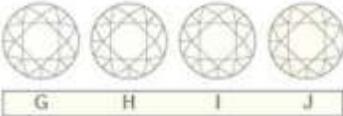
```
qplot(carat, price, data = diamonds, geom = c("point", "smooth"))
```



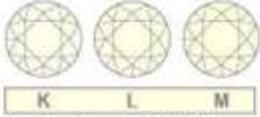
```
qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.01,  
xlim = c(0,3))
```



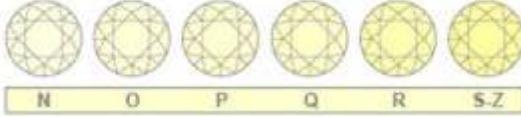
*Colorless*



*Near Colorless*

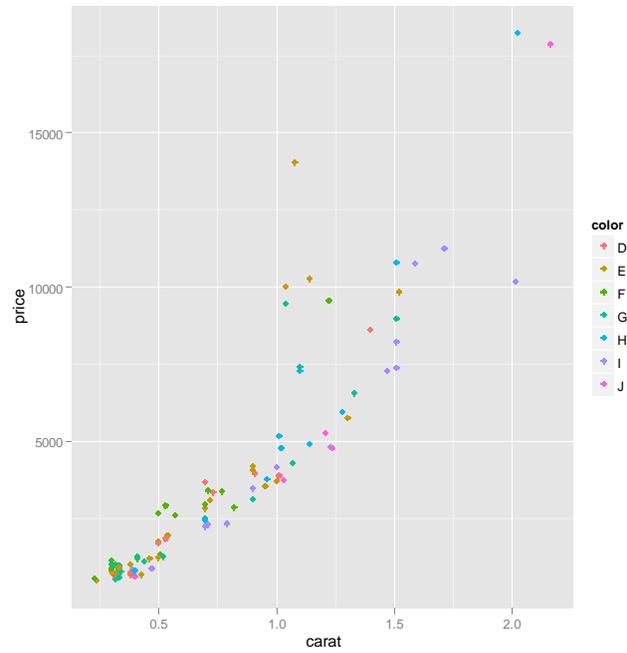


*Faint Yellow*

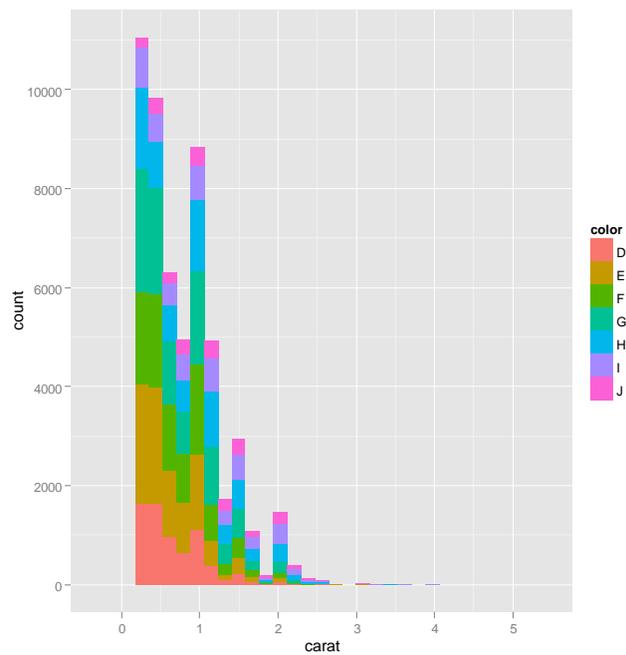


*Very Light Yellow*

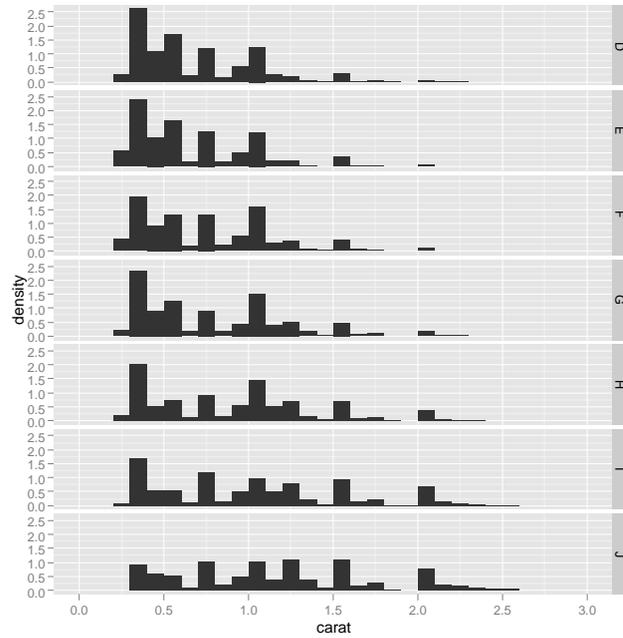
```
qplot(carat, price, colour = color,  
      data = diamonds[sample(nrow(diamonds), 100), ],)
```



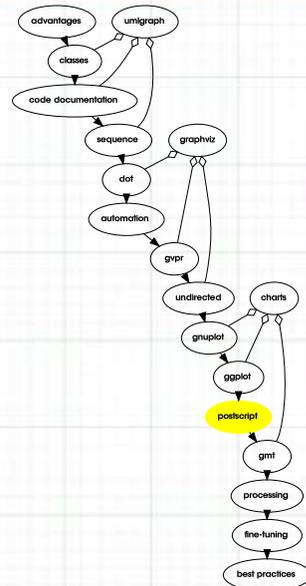
```
qplot(carat, data = diamonds, geom = "histogram", fill = color)
```



```
qplot(carat, ..density.., data = diamonds, facets = color ~ .,  
      geom = "histogram", binwidth = 0.1, xlim = c(0, 3))
```



postscript



```

grestore
} def
% Display a single ransom character
% character ransomchar -
/ransomchar {
  dup (\n) eq {
    pop
    crlf
  } {
    randfont 20 random 30 add scalefont setfont
    dup
    gsave
    /addspace false def
    5 random rotate
    3 random 1 sub 3 random 1 sub rmoveto
    dup ( ) eq {
      [ {show} ]
    } {
      [
        { show }
        { show }
        { true charpath stroke }
        { true charpath 10 random 20 div setgray fill }
        { dup stringwidth pop 0.1 mul floor 0 rmoveto
          /addspace true def
          dup box 1 setgray show }
      ]
    } ifelse
    randarray exec
    grestore
    stringwidth
    pop addspace { 1.2 mul floor } if
    0 rmoveto
    format
  } ifelse
} def
% Return a random value 0 <= r < int
% int random int
/random {
  rand
  exch
  mod
} def
% Return a random element from the array
% array randarray any
/-----

```

Change this  
note to suit your  
needs or we will  
break your legs.



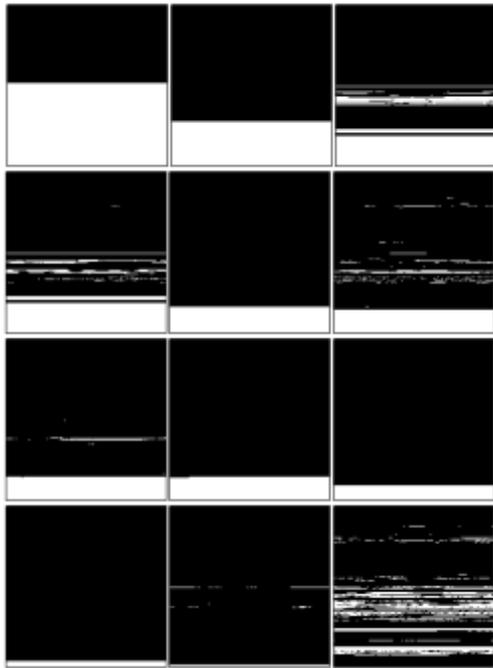
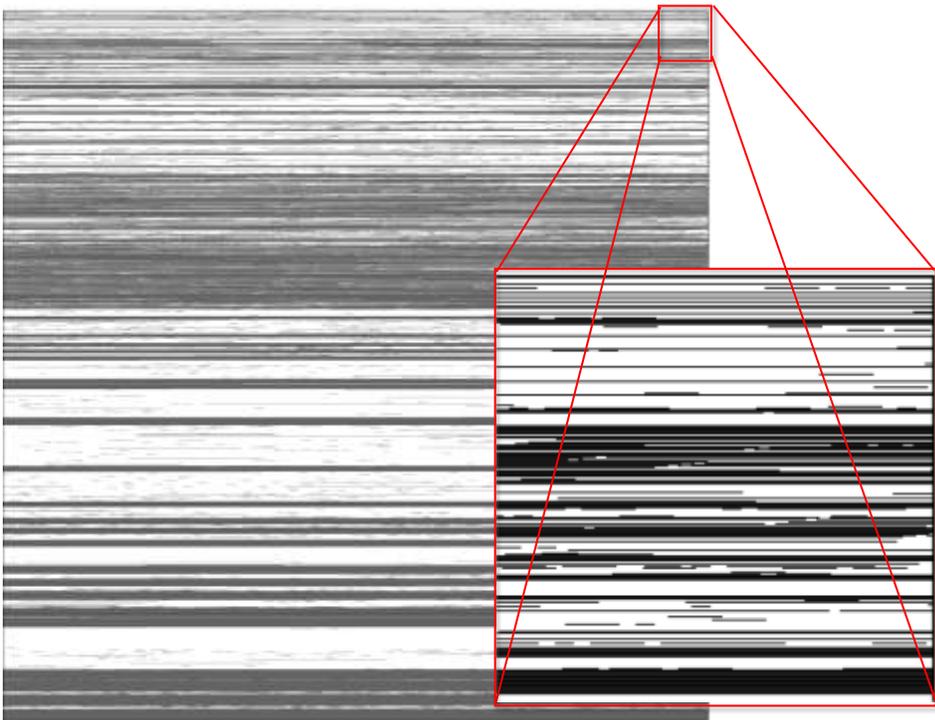


Figure 1.12: Memory pool snapshots illustrating memory fragmentation.



```

static FILE *f;
void* operator new (size_t size)
{
    if (!f)
        f = fopen("monitor.out", "w");
    void *p = malloc(size);
    if (p == 0)
        throw std::bad_alloc();
    fprintf(f, "N %u %p\n", size, p);
    return p;
}
void operator delete (void *p)
{
    free(p);
    fprintf(f, "D %p\n", p);
}

```

```

# Find range
$min = 1e9;
$max = 0;
$count = 0;
while (<IN>) {
    last if ($count++ == $snaptime);
    chop;
    if (/^N \d+ (.*)/) {
        $p = hex($1);
        $max = $p if ($p > $max);
        $min = $p if ($p < $min);
    }
}
$range = $max - $min unless (defined($range));

```

```

# Populate pool map
rewind(IN);
$count = 0;
while (<IN>) {
    last if ($count++ == $snaptime);
    chop;
    if (/^N (\d+) (.*)/) {
        $size = $1;
        $p = hex($2) - $min;
        for ($i = int($p / $range * $poolsize);
            $i < ($p + $size) / $range * $poolsize; $i++) {
            $pool[$i] = 1;
        }
        $size{$p} = $size;
    } elsif (/^D (.*)/) {
        $p = hex($1) - $min;
        $size = $size{$p};
        for ($i = int($p / $range * $poolsize);
            $i < ($p + $size) / $range * $poolsize; $i++) {
            $pool[$i] = 0;
        }
    }
}
}

```

```

prologue();
$pool[0] = 0;
$pool[$poolsize - 1] = 0;
for ($i = 1; $i < $poolsize; $i++) {
    # print STDERR "$pool[$i]";
    if ($pool[$i] && !$pool[$i - 1]) {
        # On transition
        $start = $i; # inclusive
    } elsif (!$pool[$i] && $pool[$i - 1]) {
        # Off transition; draw
        $stop = $i; # exclusive
        draw($start, $stop);
    }
}
}
epilogue();

```

```

sub draw {
    my($start, $stop) = @_;
    my($i);
    my($xstart, $ystart);
    my($xstop, $ystop);
    $ystart = int($start / $cols);
    $xstart = $start % $cols;
    $ystop = int($stop / $cols);
    $xstop = $stop % $cols;
    $ystart *= $spacing;
    $ystop *= $spacing;
    if ($ystart == $ystop) {
        print "$xstop $ystop $xstart $ystart \n";
    } else {
        for ($i = $ystart; $i <= $ystop; $i += $spacing) {
            if ($i == $ystart) {
                print "$cols $ystart $xstart $ystart \n";
            } elsif ($i == $ystop) {
                print "$xstop $ystop 0 $ystop \n";
            } else {
                print "$cols $i 0 $i \n";
            }
        }
    }
}

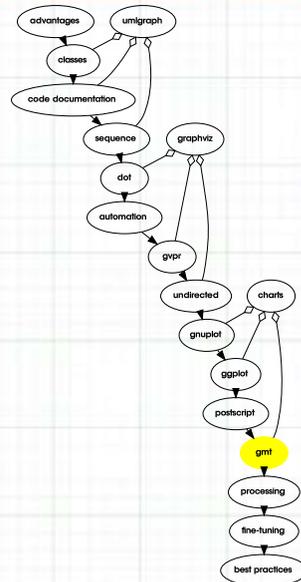
```

```

%!PS-Adobe-2.0
%%Title: Memory map
%%Pages: (atend)
%%BoundingBox: 0 0 500 500
%%EndComments
%%BeginProlog
save
/MemMapDict 200 dict def
MemMapDict begin
% Draw a line
% x1 y1 x2 y2 l -
/l {
    newpath
    moveto
    lineto
    closepath
    stroke
} def
mark
%%EndProlog
%%Page: 1 1
%%PageBoundingBox: 0 0 2568 2568
%%PageOrientation: Portrait
gsave
0 0 0 setrgbcolor
0 500 translate
0.194704049844237 -0.194704049844237 scale
2519 0 871 0 1
2568 0 2526 0 1
463 1 0 1 1
975 1 470 1 1
% [... 125,000 more lines]
% [... 125,000 more lines]
499 1860 462 1860 1
547 1860 510 1860 1
595 1860 558 1860 1
643 1860 606 1860 1
691 1860 654 1860 1
739 1860 702 1860 1
811 1860 774 1860 1
% Draw the border
2 setlinewidth
0 0 moveto
0 2568 lineto
2568 2568 lineto
2568 0 lineto
0 0 lineto
stroke
grestore
cleartomark
%%PageTrailer
%%EndPage: 1
%%Trailer
%%Pages: 1
end
restore
%%EOF

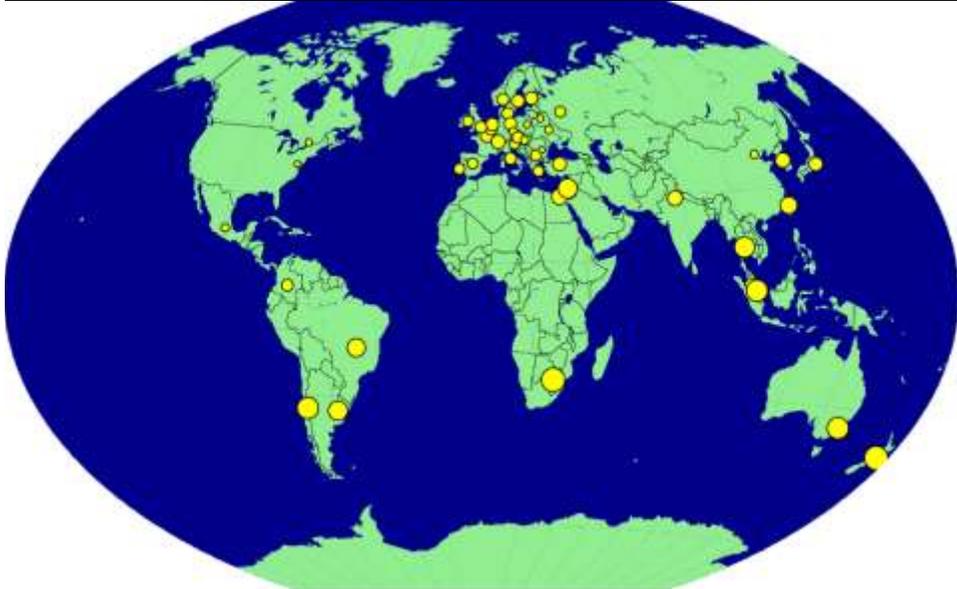
```

# GMT: generic mapping tools



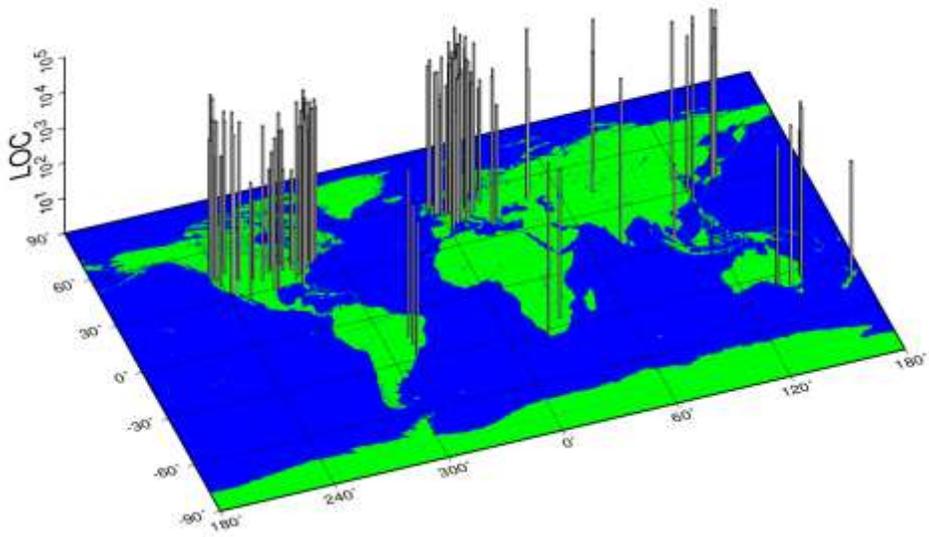
```
# -R Global domain
# -JR0/9i Winkel Tripel projection at 0 9"
# -Dc crude resolution
# -A5000 omit features less than 5000km
# -N1 Draw country political boundaries
# -K Expect more output
pscoast -Rg -JR0/15c -Glightgreen -Sdarkblue \
  -Dc -A5000 -N1 -K >\map.eps
```

```
# -O Overlay
# -R Global domain
# -JR0/9i Winkel Tripel projection at 0 9"
# -Sc Draw circle in cm
# -G fill color
# -W pen width
psxy -O -Rg -JR0/15c -Sc -Gyellow -Wthinner \
  author-country-co2-loc.txt >>map.eps
```

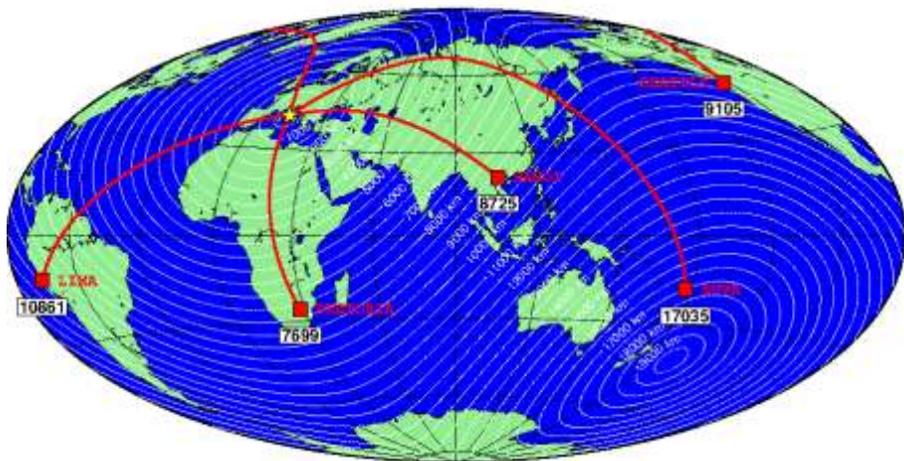


```
pscoast -R-180/180/-90/90 -JX8i/5id -Dc -G0 \  
-E200/40 -K W0.25p/255/255/255 -G0/255/0 \  
-S0/0/255 -Di -P >map.eps
```

```
psxyz loc-lines.dat -P -R-180/180/-90/90/1/100000 \  
-JX -JZ2.5il -So0.02ib1 -G140 -W0.5p -O -E200/40 \  
-B60g60/30g30/a1p:LOC:WSneZ >> map.eps
```

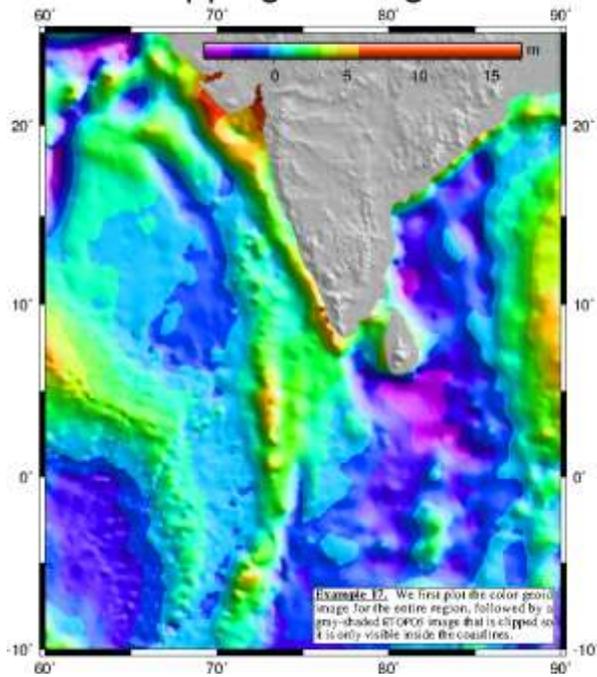


## Distances from Rome to the World



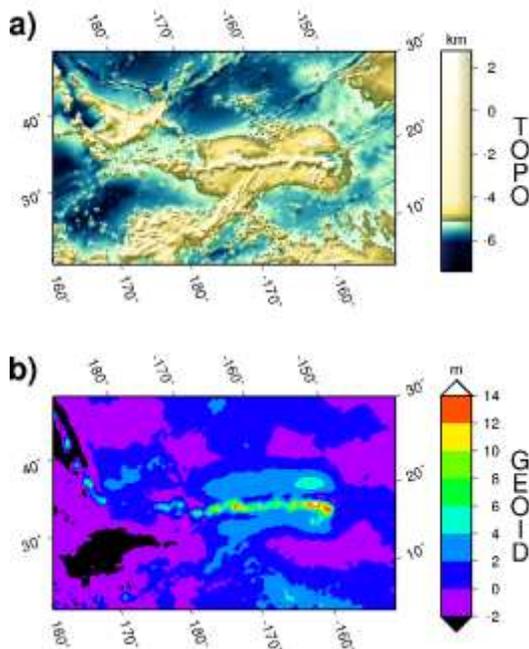
grdmath pscost grdcontour psxy pstext grdtrack

## Clipping of Images



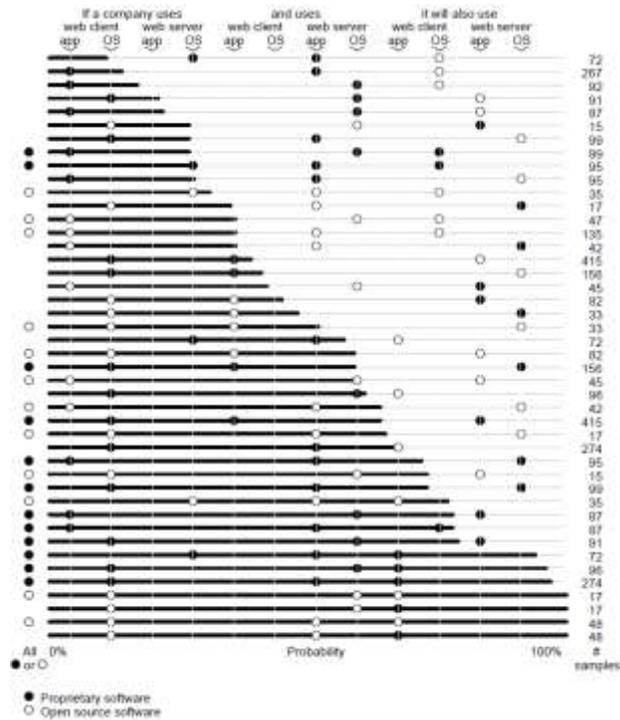
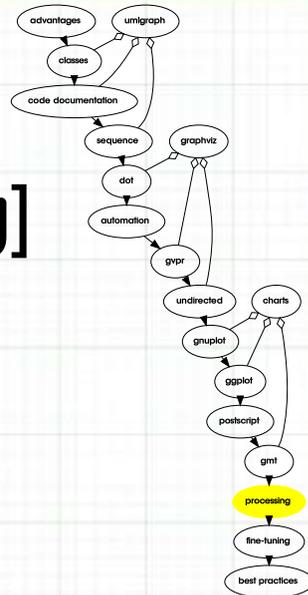
grd2cpt  
 grdgradient  
 grdimage  
 pscost  
 grdgradient  
 grdimage  
 pscost  
 psscale  
 pstext

## HAWAIIAN TOPO AND GEOID



makecpt  
 grdimage  
 psscale  
 grd2cpt  
 grdgradient  
 grdimage  
 psscale  
 pstext

# processing[.org]



```

import processing.pdf.*;
PFont arial;
boolean pdf = true;
int NENTRIES = 44;
int columnWidth = 20;
int radius = 8;
int left = 40;
int leading = 13;
int OSSbrowser = 0;
int OSSClientOS = 1;
int OSSWebServ = 2;
int OSSServOS = 3;
int Propbrowser = 4;
int PropClientOS = 5;
int PropWebServ = 6;
int PropServOS = 7;
boolean isProp(int point) {
    return point > 3;
}
void drawPoint(int point, int x, int y) {
    fill(isProp(point) ? 0 : 255);
    ellipse(left + x + columnWidth * (point % 4) * 2, y, radius, radius
}

```

```

class Entry {
    int uses1, uses2;
    int willUse;
    int probability;
    int samples;
    Entry(int u1, int u2, int w, int p, int s) {
        uses1 = u1;
        uses2 = u2;
        willUse = w;
        probability = p;
        samples = s;
    }
}

```

```

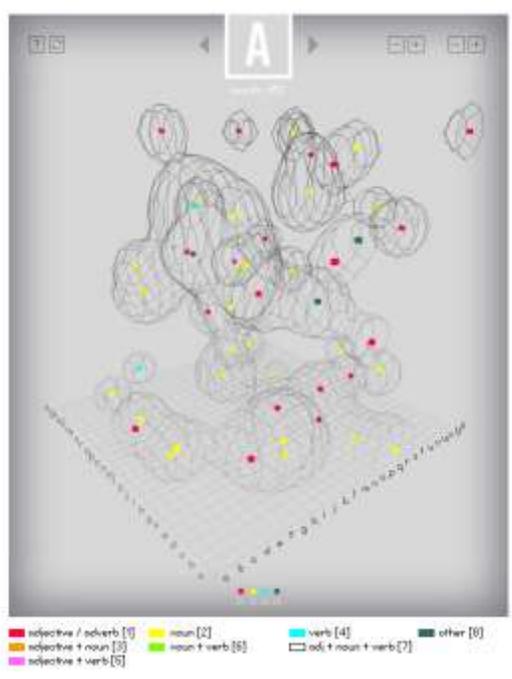
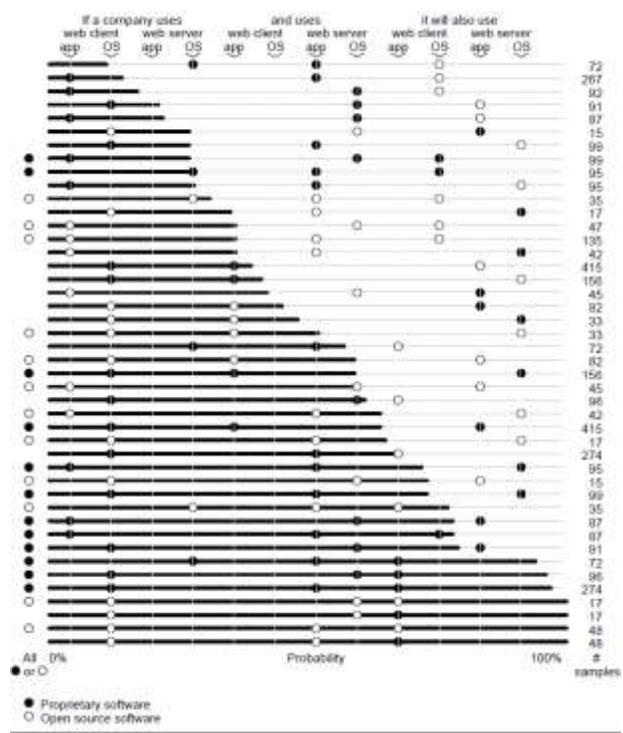
void draw(int y) {
    stroke(190);
    line(left, y, width * .9, y);
    stroke(0);
    strokeWeight(3);
    line(left, y, left + probability /
        100.0 * (width - left) * .9, y);
    strokeWeight(1);
    fill(0);
    text(samples, width * .96 + 3, y);
    drawPoint(uses1, columnWidth, y);
    drawPoint(uses2, 9 * columnWidth, y);
    drawPoint(willUse, 17 * columnWidth, y);
    if (isProp(uses1) == isProp(uses2) &&
        isProp(uses2) == isProp(willUse)) {
        fill(isProp(uses1) ? 0 : 255);
        ellipse(left / 2, y, radius, radius);
    }
}
};

```

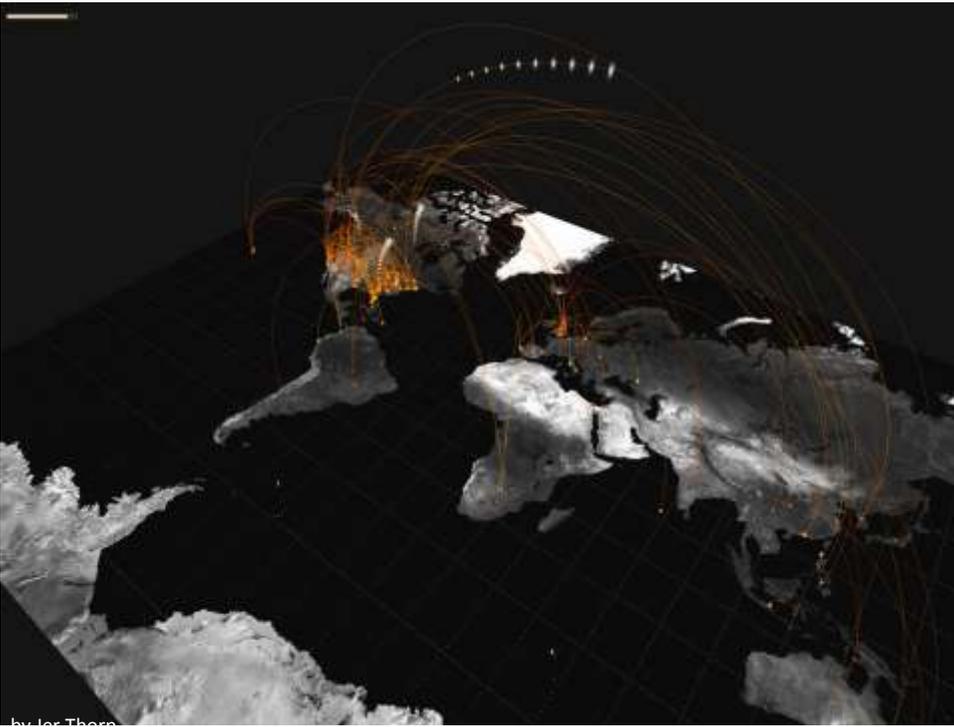
```

stroke(0);
// Draw bottom legend
int y = (5 + NENTRIES) * leading - leading / 2;
textAlign(CENTER);
text("All", left / 2, y);
text("or", left / 2, y + leading);
ellipse(left / 2 - radius - 5, y + radius, radius, radius);
fill(255);
ellipse(left / 2 + radius + 5, y + radius, radius, radius);
fill(0);
textAlign(LEFT);
text("0%", left, y);
text("#", width * .95, y);
text("samples", width * .92, y + leading);
textAlign(RIGHT);
text("100%", width * .9, y);
textAlign(CENTER);
text("Probability", width / 2, y);
y += 3 * leading;
textAlign(LEFT, CENTER);
fill(0);
ellipse(left / 2, y, radius, radius);
text(" Proprietary software", left / 2 + radius, y);
y += leading;
fill(255);
ellipse(left / 2, y, radius, radius);
fill(0);
text(" Open source software", left / 2 + radius, y);
println(y);
noLoop();
}

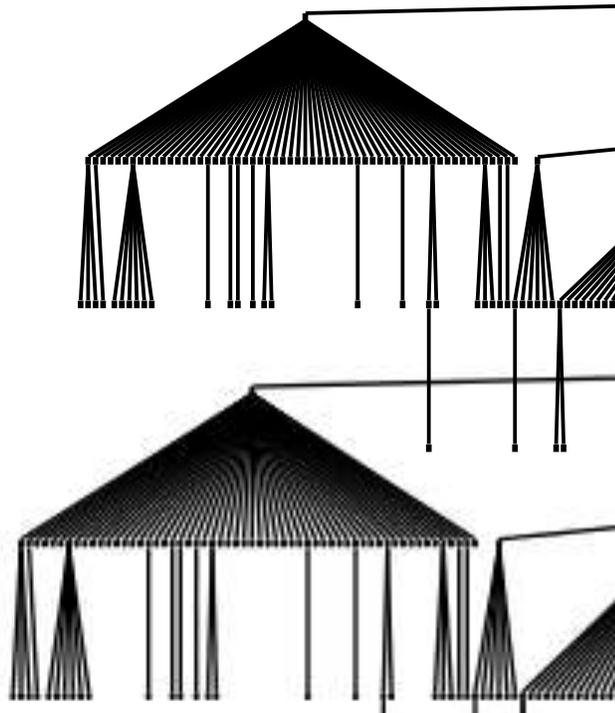
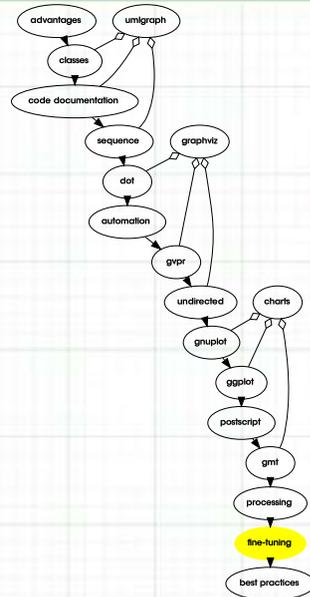
```

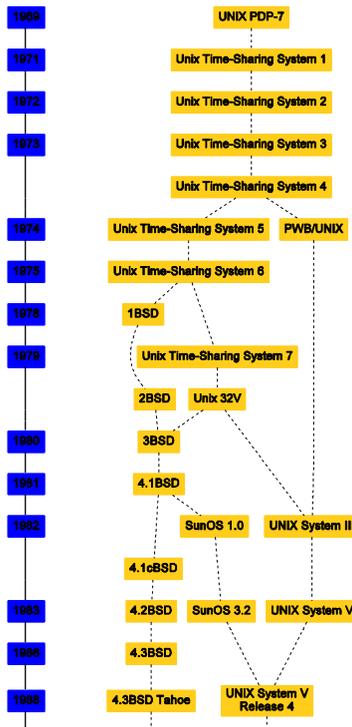
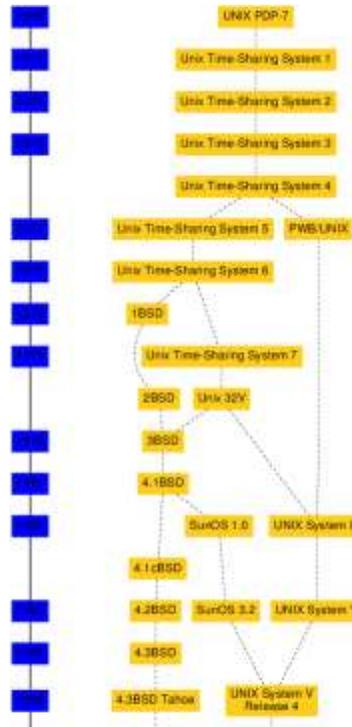
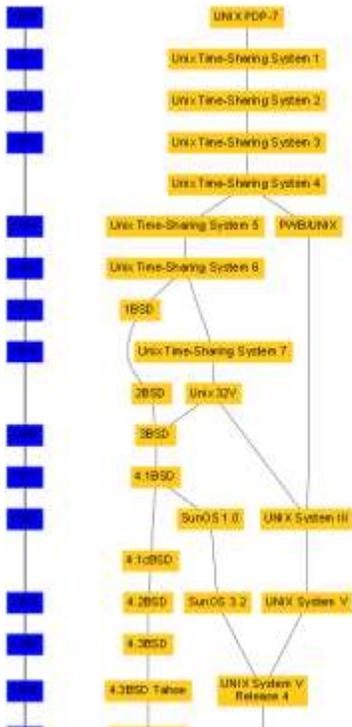


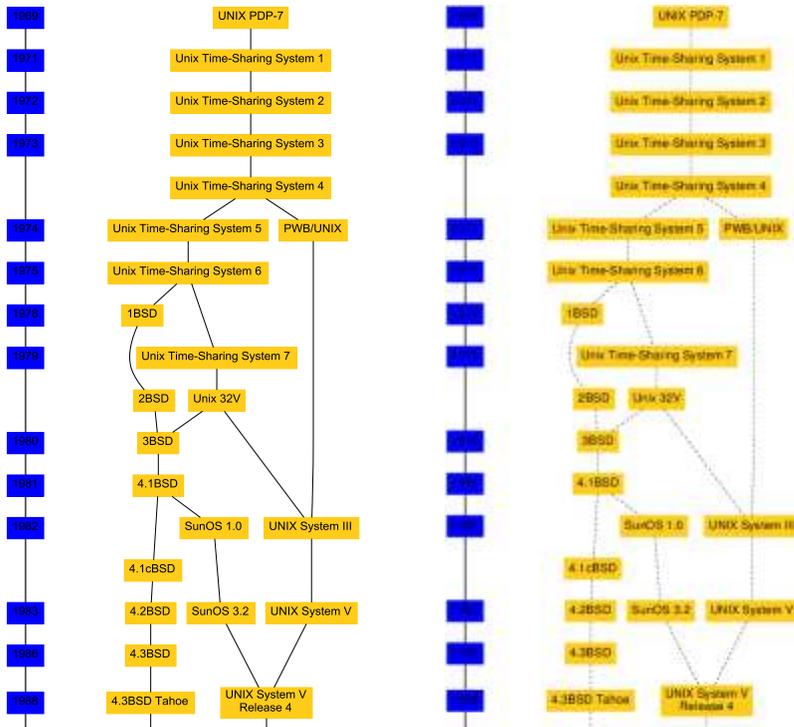
By Toxi



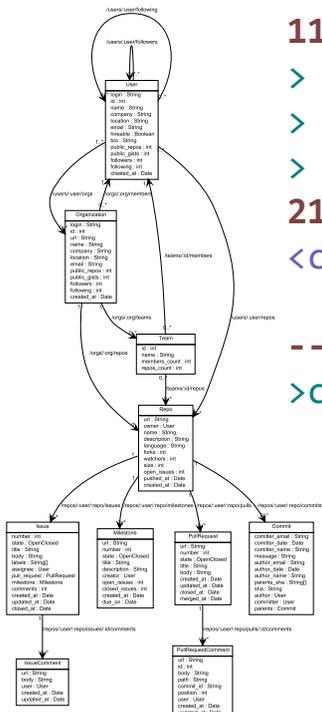
# fine-tuning







hand tuning



11a12,14

- > {rank=source; c4; }
- > {rank=same; c5; c6; }
- > {rank=same; c7; c2; c8; }

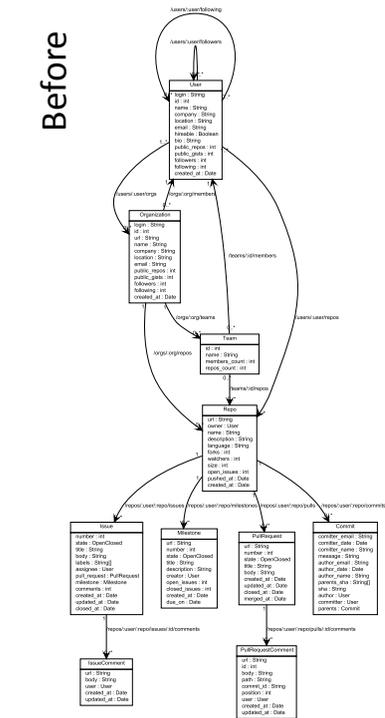
210c213

<c7:p -> c2:p [taillabel="1", headlabel="0..\*" ...];

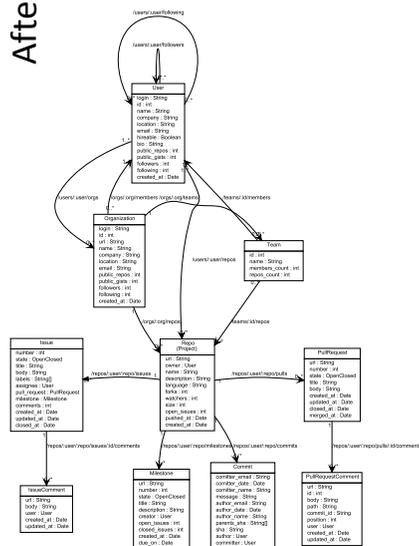
---

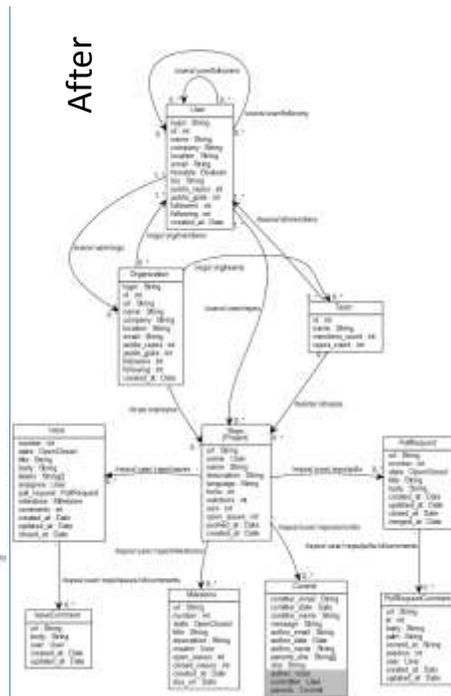
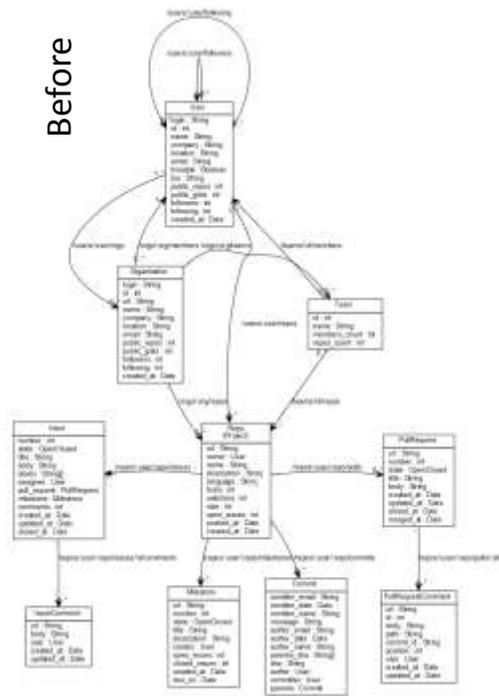
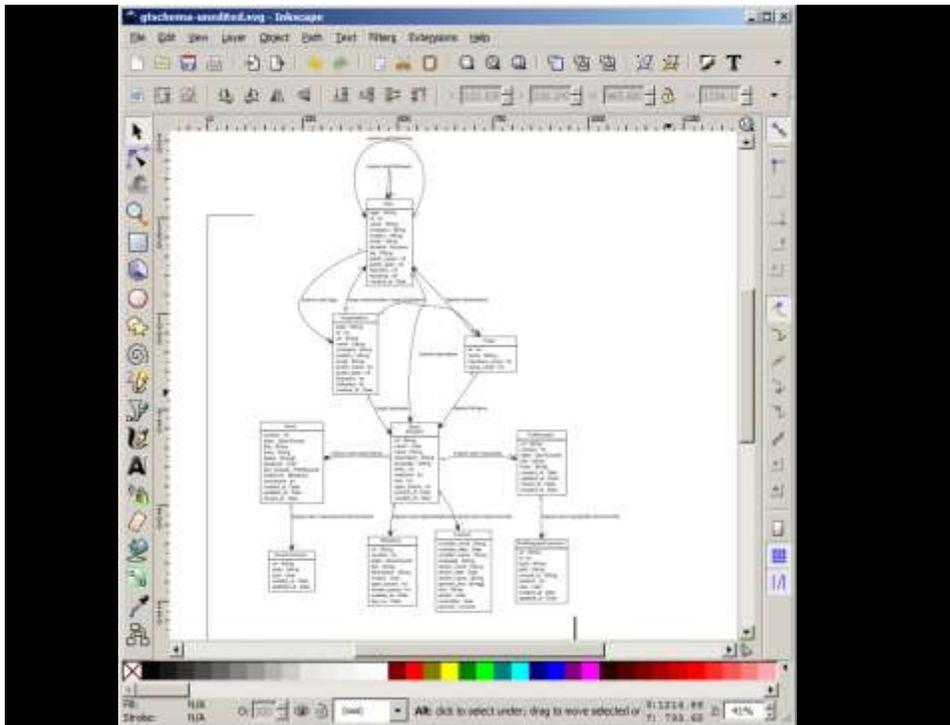
>c2:p -> c7:p [headlabel="1", dir=back, taillabel="0..\*" ...];

Before

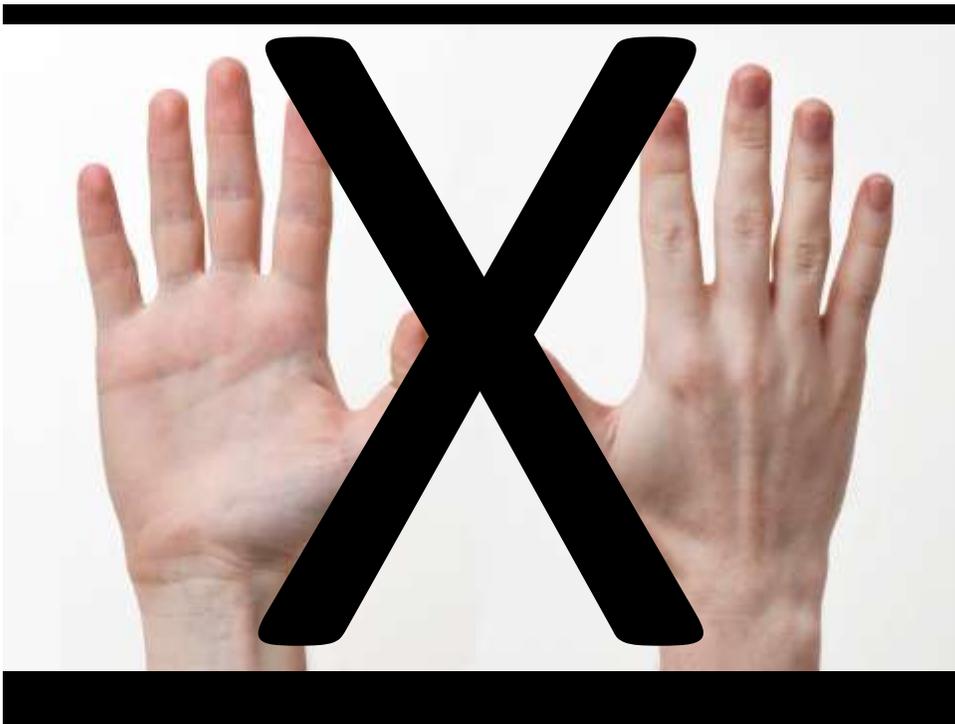
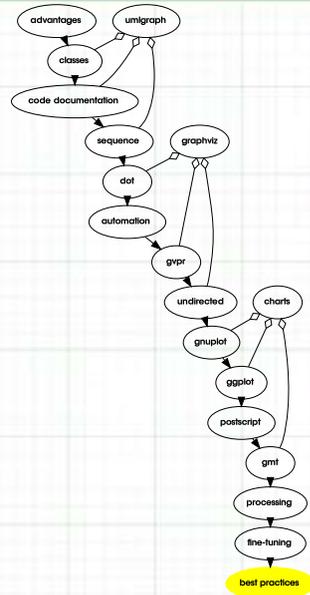


After

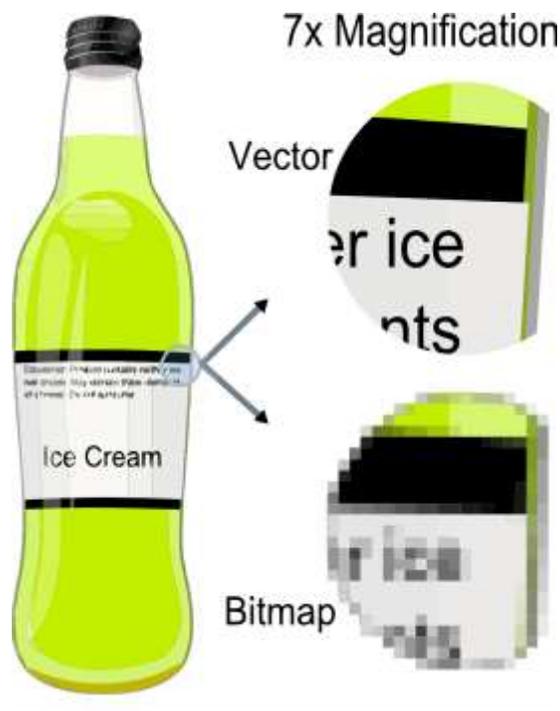




# best practices





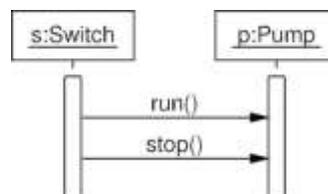


```

# Define the objects
object(S, "s:Switch");
object(P, "p:Pump");
step();
active(S);
active(P);

# Message sequences
message(S, P, "run()");
message(S, P, "stop()");
step();
complete(S);
complete(P);

```





[www.spinellis.gr](http://www.spinellis.gr)  
[github.com/dspinellis](https://github.com/dspinellis)  
[twitter.com/coolsweng](https://twitter.com/coolsweng)  
[dds@aub.gr](mailto:dds@aub.gr)

