Software Product Lines

Part 3: Versioning, build systems, and preprocessors

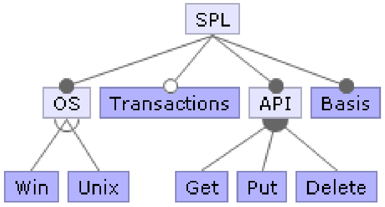
Daniel Strüber, Radboud University

with courtesy of: **Sven Apel, Christian Kästner, Gunter Saake**

How to implement variability?

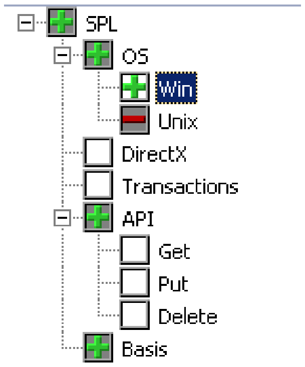
Domain Eng.

Feature model



Reusable implementation artifacts

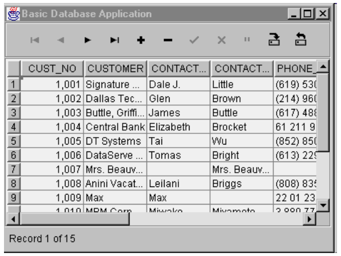
Application Eng.



- ☑ SPL
  - ☑ OS
    - ☑ Win
    - ☒ Unix
  - □ DirectX
  - □ Transactions
  - ☑ API
    - □ Get
    - □ Put
    - □ Delete
  - ☑ Basis

Feature selection

Generator



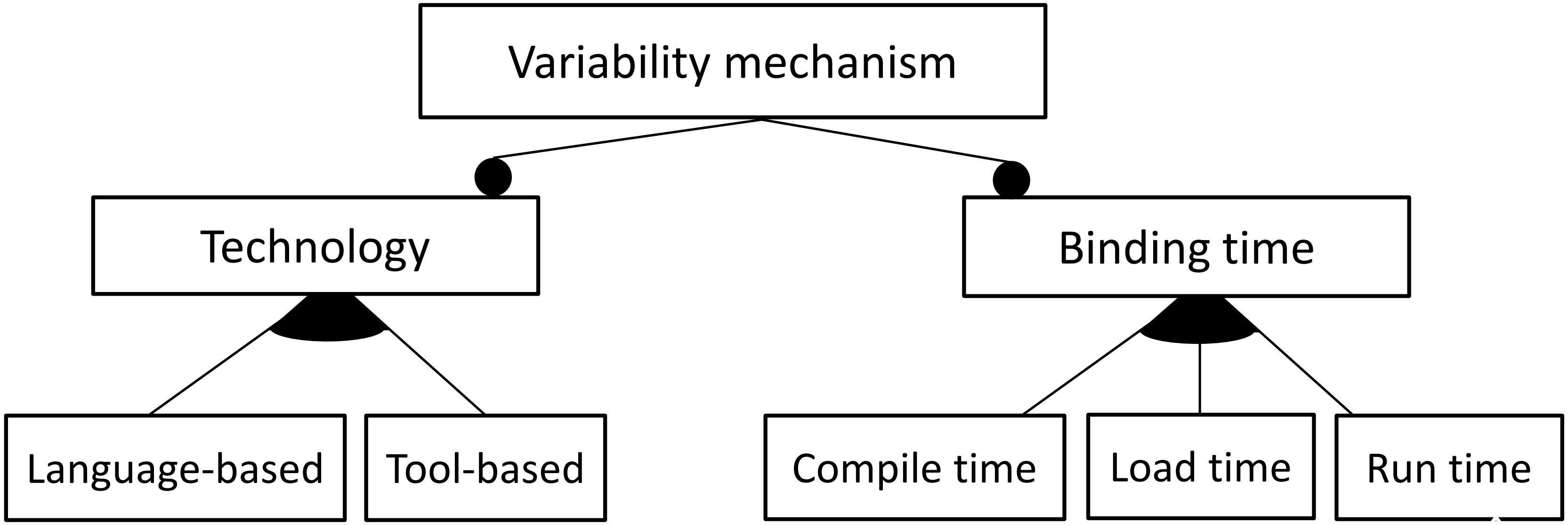
Basic Database Application

	CUST_NO	CUSTOMER	CONTACT...	CONTACT...	PHONE_
1	1,001	Signature ...	Dale J.	Little	(619) 53
2	1,002	Dallas Tec...	Glen	Brown	(214) 96
3	1,003	Buttle, Griffi...	James	Buttle	(617) 48
4	1,004	Central Bank	Elizabeth	Brocket	61 211 9
5	1,005	DT Systems	Tai	Wu	(852) 85
6	1,006	DataServe ...	Tomas	Bright	(613) 22
7	1,007	Mrs. Beauv...		Mrs. Beauv...	
8	1,008	Anini Vacat...	Leilani	Briggs	(808) 83
9	1,009	Max	Max		22 01 23
	1,010	MDM Corp	Miwako	Miyamoto	3 880 77

Record 1 of 15

Final program

Variability mechanisms: a broad categorization



Variability at *compile time*

- *Goal*: only compile the source code required for current product
  - Smaller, optimised variants
  - Source code selected, compiled and packaged appropriately

- How to implement options or alternatives?
- For now: simple means for a few variants

Version control systems

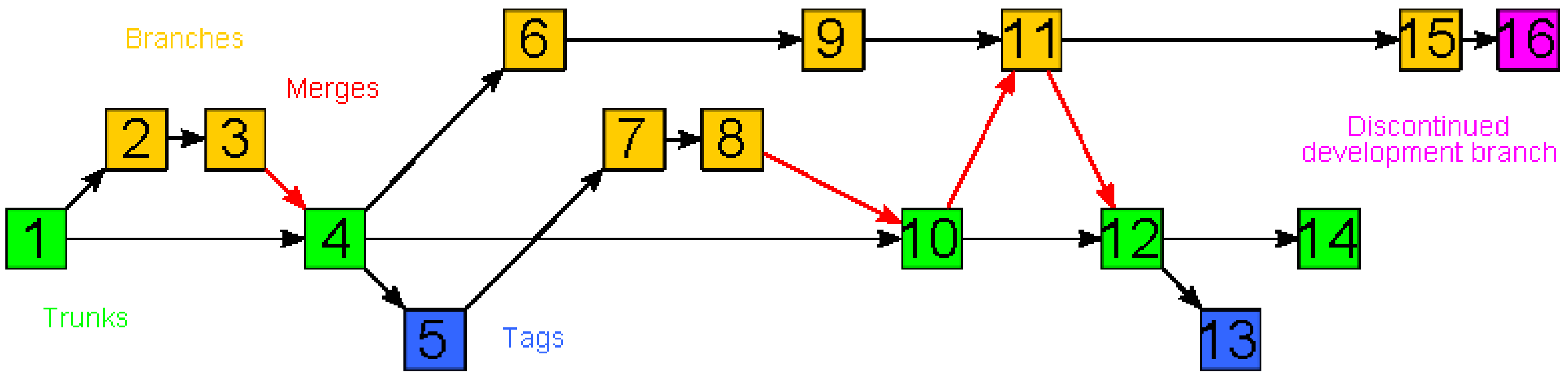
Version control systems

- Support versioning of (typically source-code) files
  - Collaborative development
  - Archive of old file versions
    - Time stamps and user (author) names
    - Changes typically stored as delta
  - Process: Checkout – Change – Commit – Update – Change – Commit - …
- Example systems: Git, Mercurial, SVN , CVS, Perforce, Visual SourceSafe, SCCS
- Here used as a configuration management tool

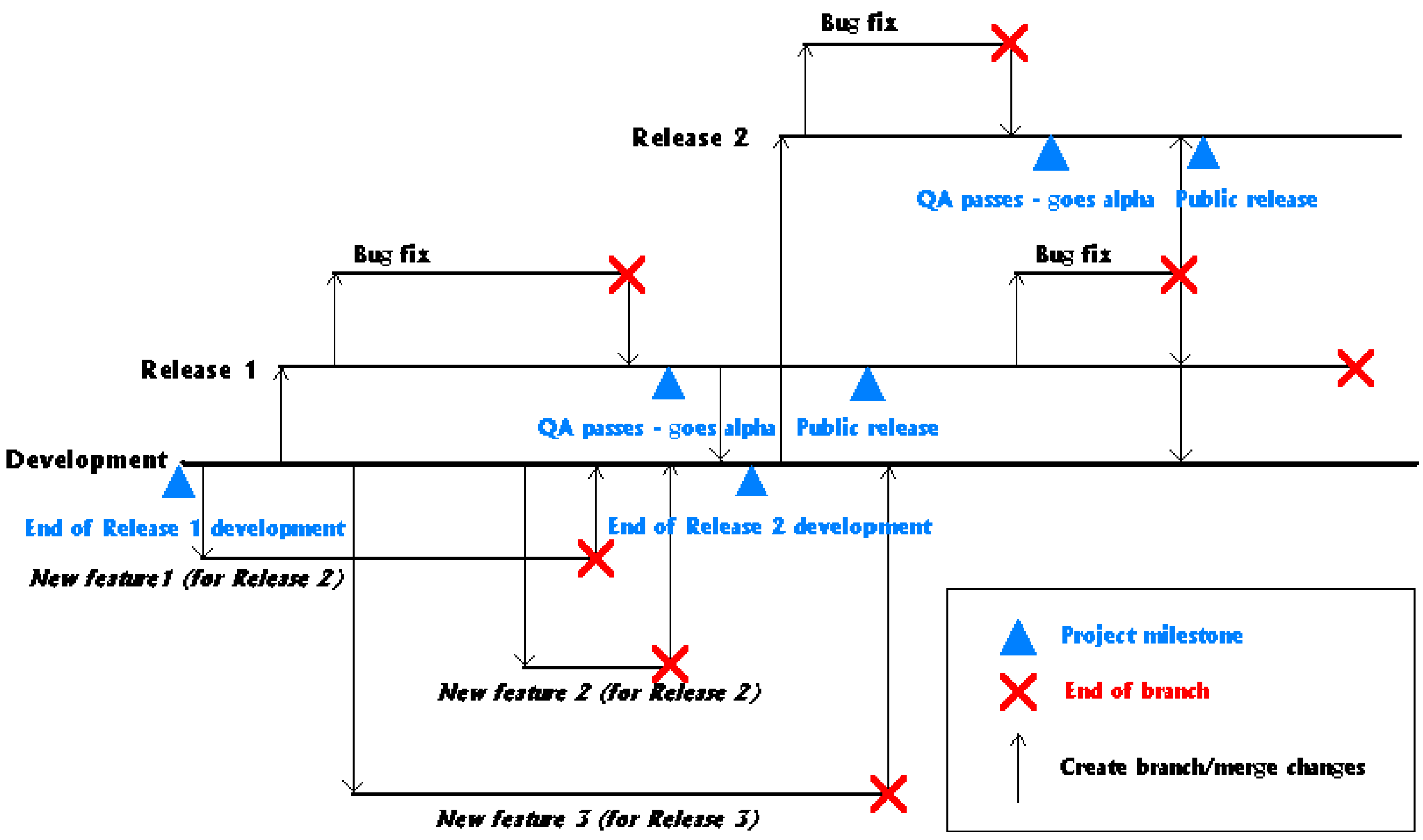
Code and non-code files

- Java code
- Documentation
- Models
- Build scripts: ant/makefile
- Licences
- Grammars
- HTML, JavaScript, CSS
- *Compiled binary files*
  - *Differencing, conflict management and merging more difficult*

Branching & Merging

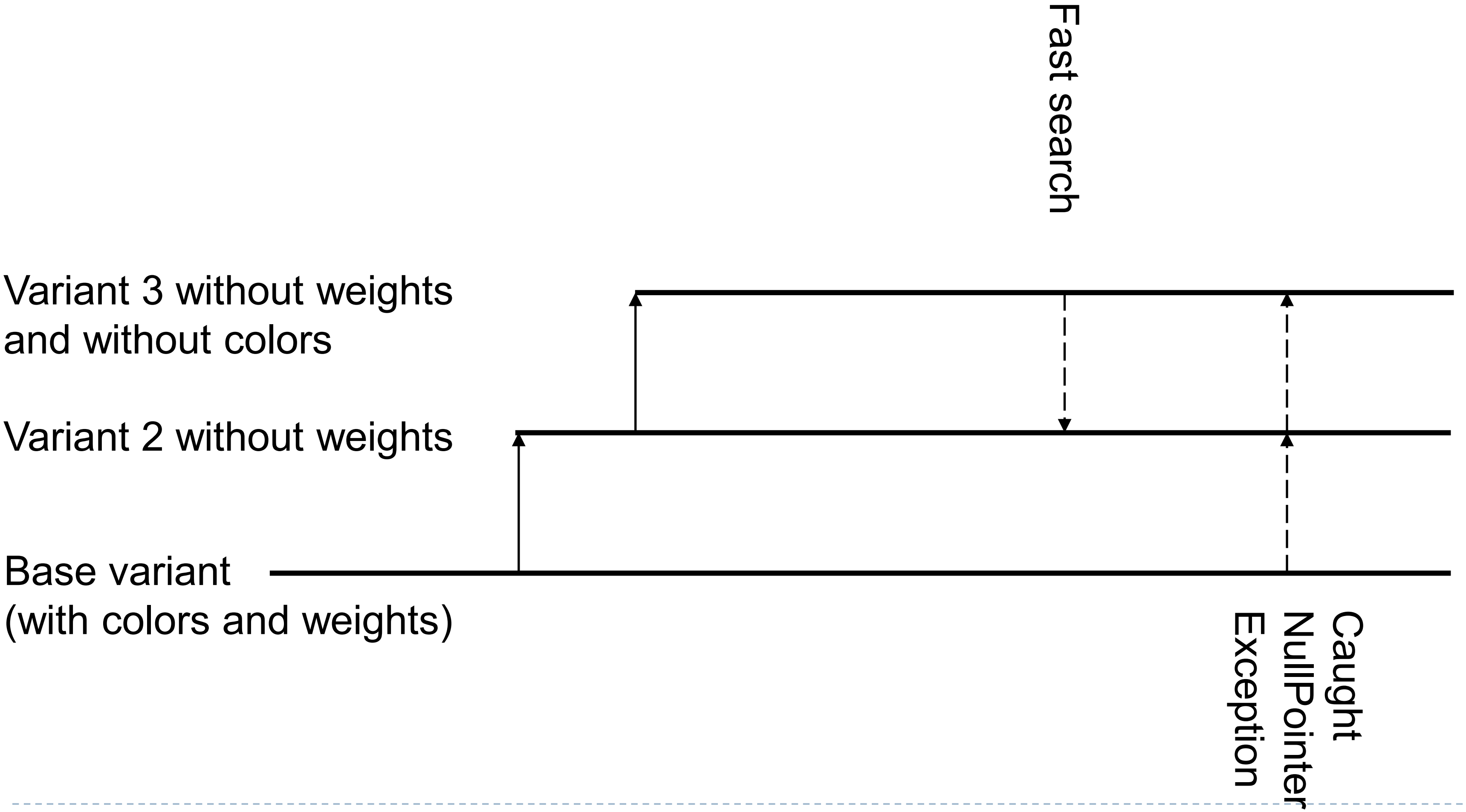


Release planning (Common case)


Bug fix
Release 2
QA passes - goes alpha
Public release
Bug fix
Bug fix
Release 1
QA passes - goes alpha
Public release
Development
End of Release 1 development
End of Release 2 development
New feature 1 (for Release 2)
New feature 2 (for Release 2)
New feature 3 (for Release 3)
Project milestone
End of branch
Create branch/merge changes

Branching and merging during development and releases

Release planning: Graph library



Fast search

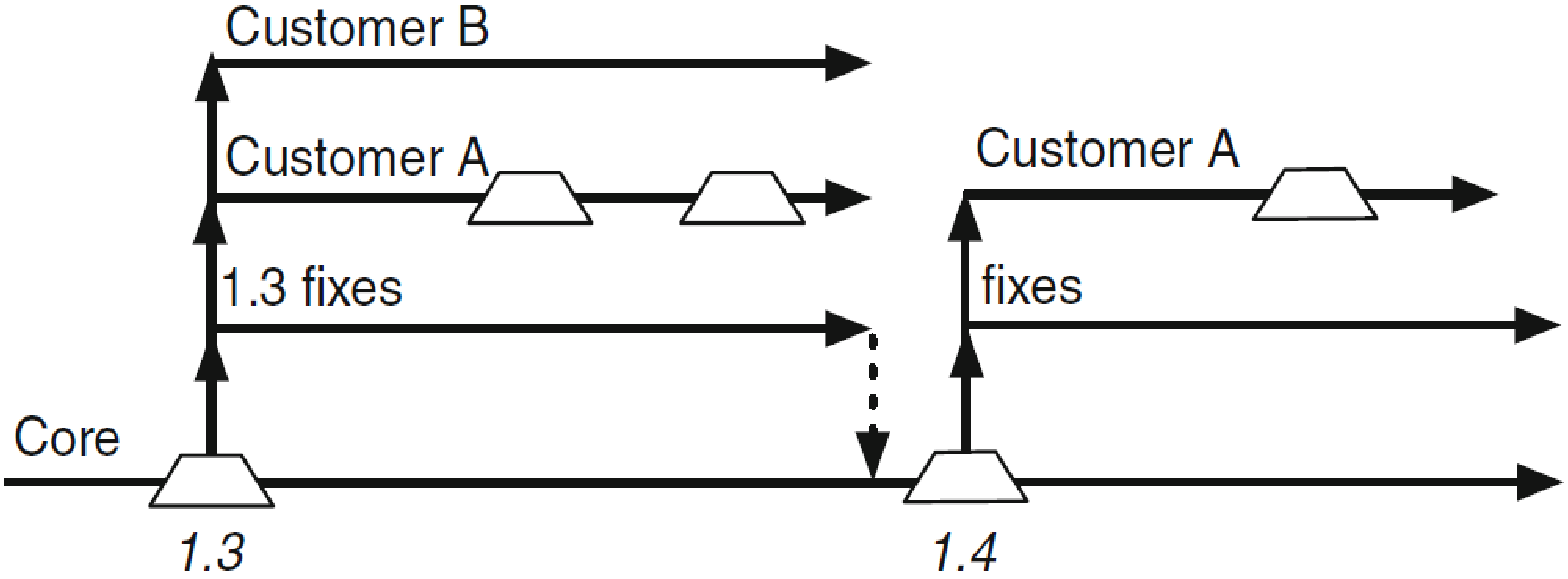
Variant 3 without weights and without colors

Variant 2 without weights

Base variant (with colors and weights)

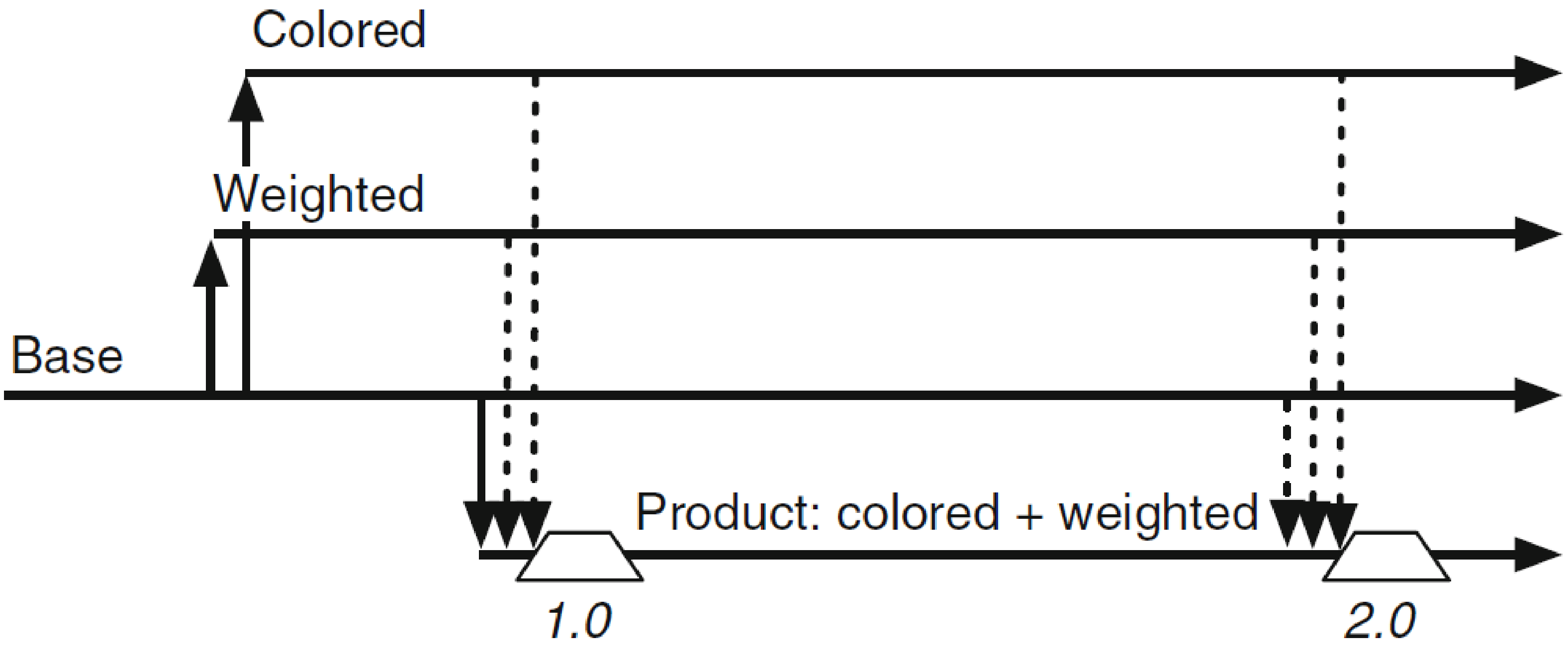
Caught NullPointer Exception

Alternative release plannings: Customer-specific



Customer-specific variations: a branch per customer

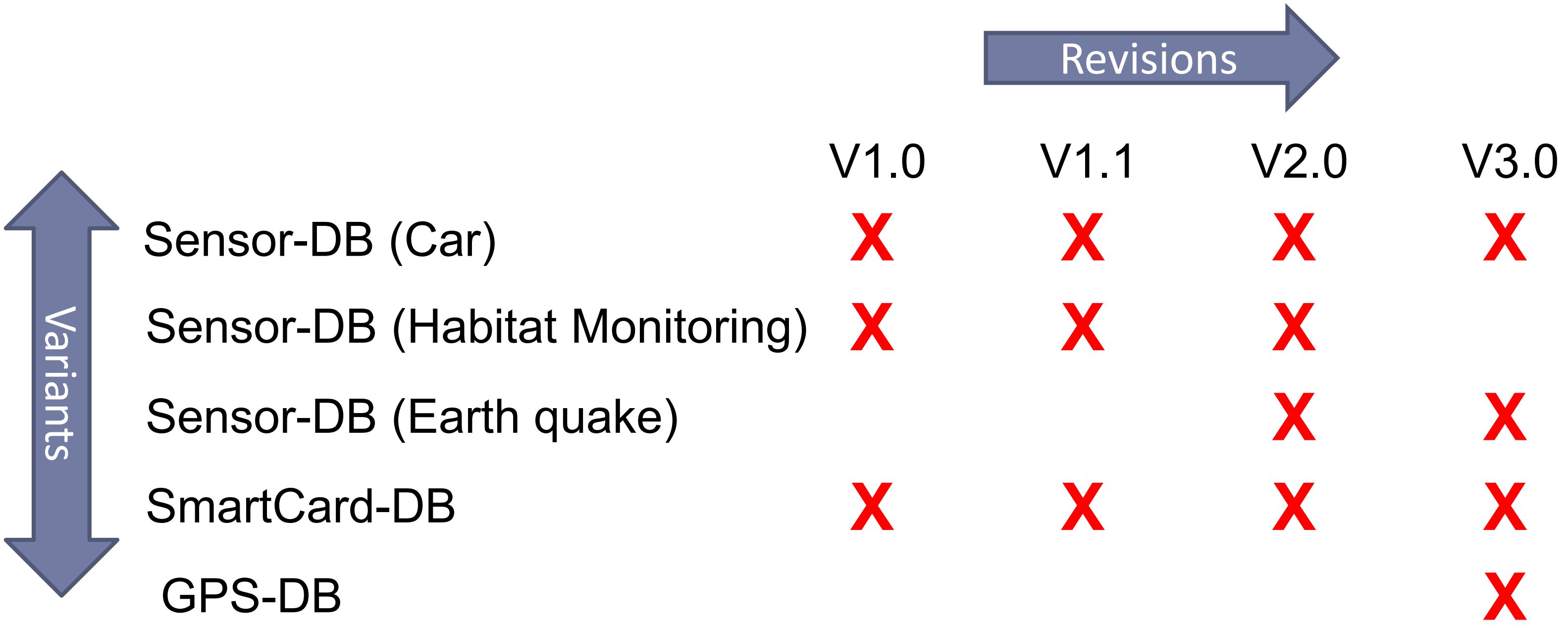
Alternative release plannings: Feature-oriented



Feature-oriented variations: a branch per feature

Variants vs. Revisions vs. Versions

- **Revisions**: snapshots at a particular point *in time*, with identifier and (optionally) description; ordered.
- **Variants**: versions of the same artifact existing in parallel; coexist in space, usually not ordered.
- **Version**: umbrella term for revisions and variants



	V1.0	V1.1	V2.0	V3.0
Sensor-DB (Car)	X	X	X	X
Sensor-DB (Habitat Monitoring)	X	X	X	
Sensor-DB (Earth quake)			X	X
SmartCard-DB	X	X	X	X
GPS-DB				X

Product lines with version control systems

- Development of variants in branches
- On changes, might need to synchronize between branches

Domain Eng.

Variant list

•Sensor-DB (Car)
•Sensor-DB (Habitat Monitoring)
•Sensor-DB (Earth quake)
•SmartCard-DB
•GPS-DB

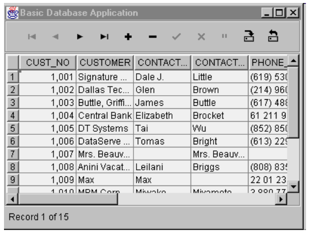
Mapping
branch <-> variant

Version control
with branches

Application Eng.

□ Sensor-DB (Car)
☑ Sensor-DB (Habitat Monitoring)
□ Sensor-DB (Earth quake)
□ SmartCard-DB
□ GPS-DB

Variant as
input



	CUST_NO	CUSTOMER	CONTACT...	CONTACT...	PHONE_
1	1,001	Signature ...	Dale J.	Little	(619) 530
2	1,002	Dallas Tec...	Glen	Brown	(214) 960
3	1,003	Buttle, Griffi...	James	Buttle	(617) 488
4	1,004	Central Bank	Elizabeth	Brocket	61 211 9
5	1,005	DT Systems	Tai	Wu	(852) 850
6	1,006	DataServe ...	Tomas	Bright	(613) 229
7	1,007	Mrs. Beauv...		Mrs. Beauv...	
8	1,008	Anini Vacat...	Leilani	Briggs	(808) 835
9	1,009	Max	Max		22 01 23
	1,010	MRM Corp	Miyako	Miyamoto	3 880 77

Record 1 of 15

Variant selection

Check out the
corresponding branch

Complete program

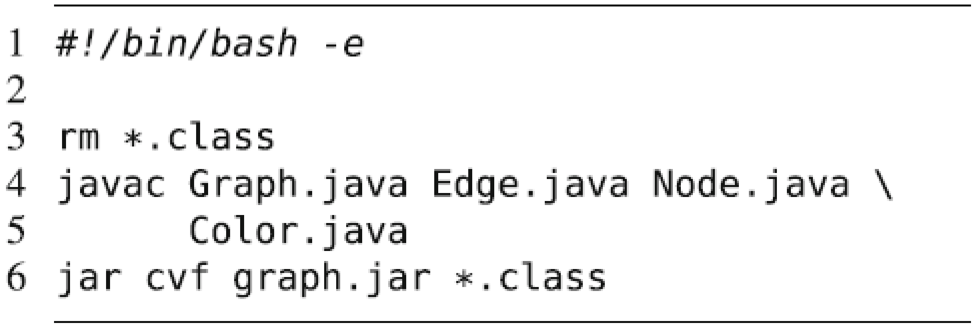
Product lines with version control systems: Discussion

- Benefits
  - Established, reliable systems
  - Familiar process
  - Good tool integration
  - Language-independent (non-code files supported)
- Drawbacks
  - Developing variants instead of features: flexible combination of features not directly possible
  - No structured reuse (clone & own)
  - High maintenance effort (bugfixes have to be propagated between branches)
  - Tool support is largely text-based instead of language-based, often wrong level of abstraction

Build systems

Build systems

- Automation of build process
  - Copy files, clean directories
  - Call compiler and additional tools (e.g., JavaDoc generator)
  - Launch the tests
- Multiple steps, with dependencies/conditions
- Tools: make, ant, maven, gradle, …



```
#!/bin/bash -e

rm *.class
javac Graph.java Edge.java Node.java \
      Color.java
jar cvf graph.jar *.class
```

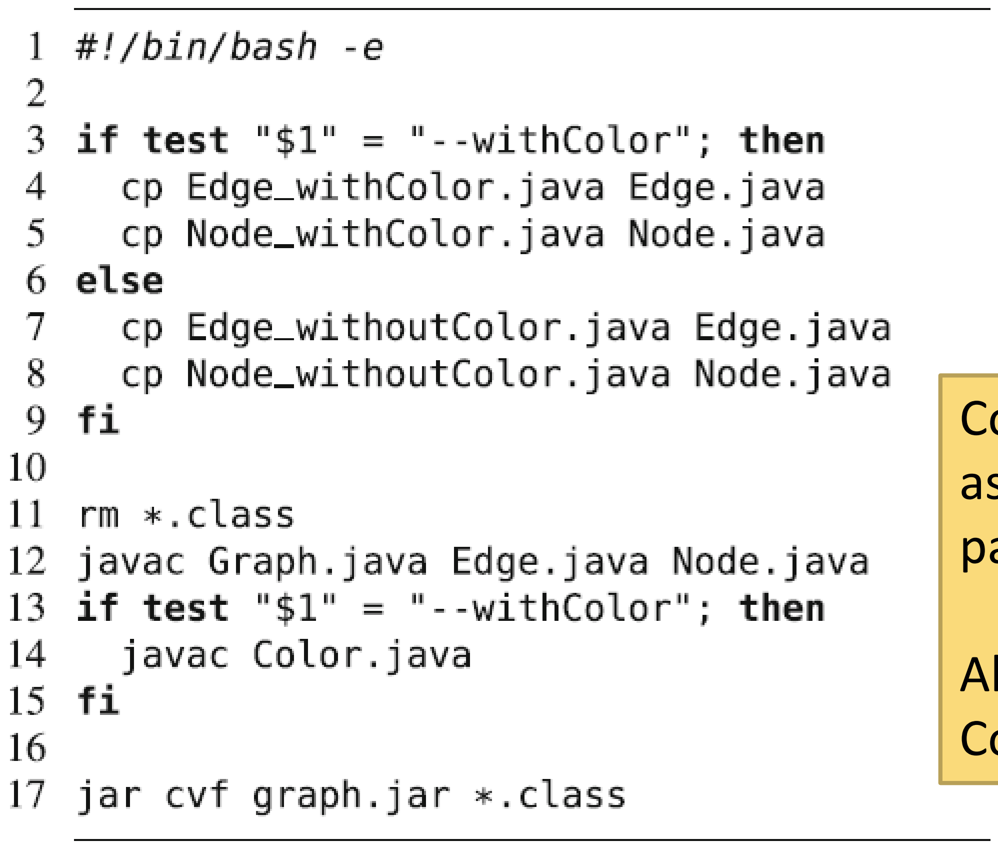
Graph library: simple build script

Professional build systems

- Make, Ant, Maven, Gradle…
  - Multiple build targets
  - Automated dependency resolution
  - Incremental builds for performance optimization
  - Preparing a build report
  - Reduction of specification effort by conventions
    (Maven: standard structure for projects and configurations)

- Decide what is compiled when and where
  - Candidate for compile time variability
  - Multiple solutions possible

Solution 1: build script + config options



```
#!/bin/bash -e

if test "$1" = "--withColor"; then
  cp Edge_withColor.java Edge.java
  cp Node_withColor.java Node.java
else
  cp Edge_withoutColor.java Edge.java
  cp Node_withoutColor.java Node.java
fi

rm *.class
javac Graph.java Edge.java Node.java
if test "$1" = "--withColor"; then
  javac Color.java
fi

jar cvf graph.jar *.class
```

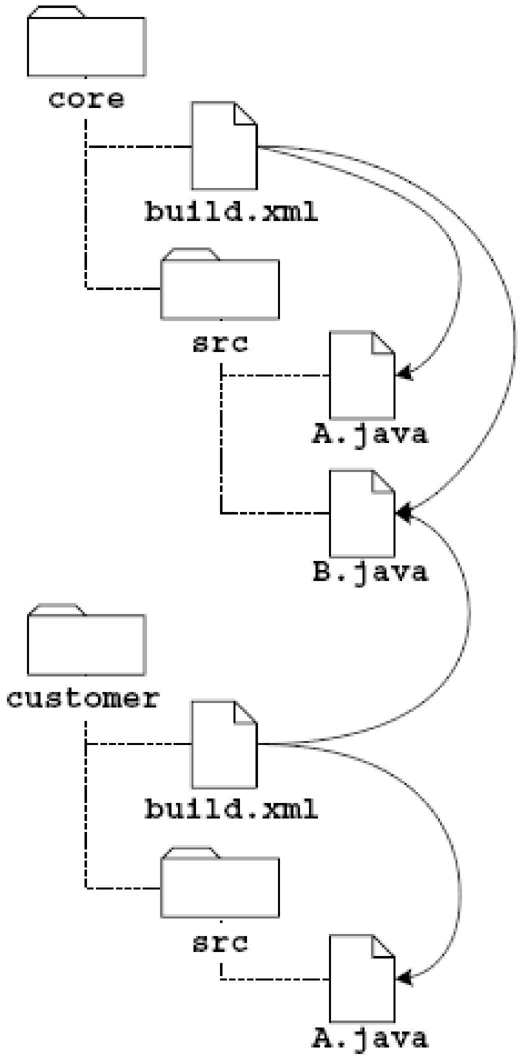
Config options here as command line parameters.

Alternative:
Config file

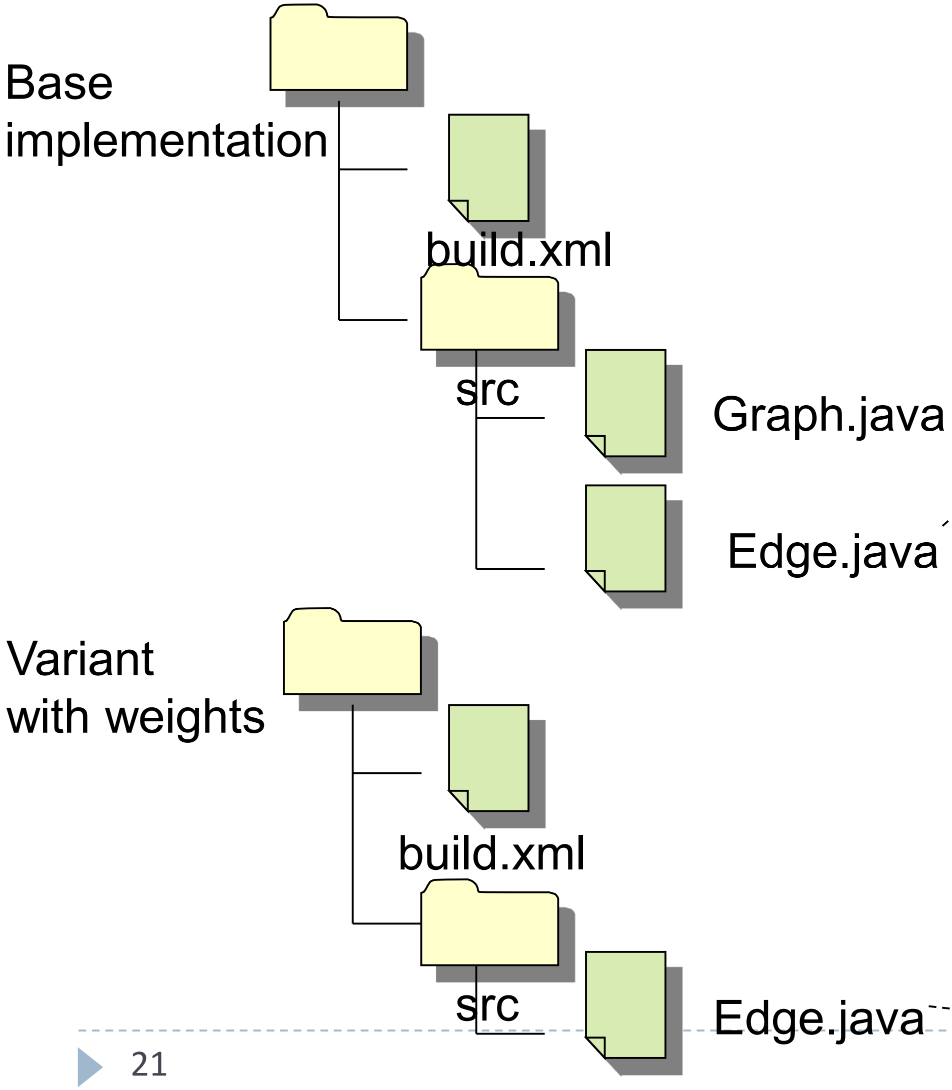
Graph library: build file enriched with variability

Solution 2: product lines with build systems

- Per variant, one config file or build script
- On compilation, include or exclude files
- Overwrite files with product-specific variants



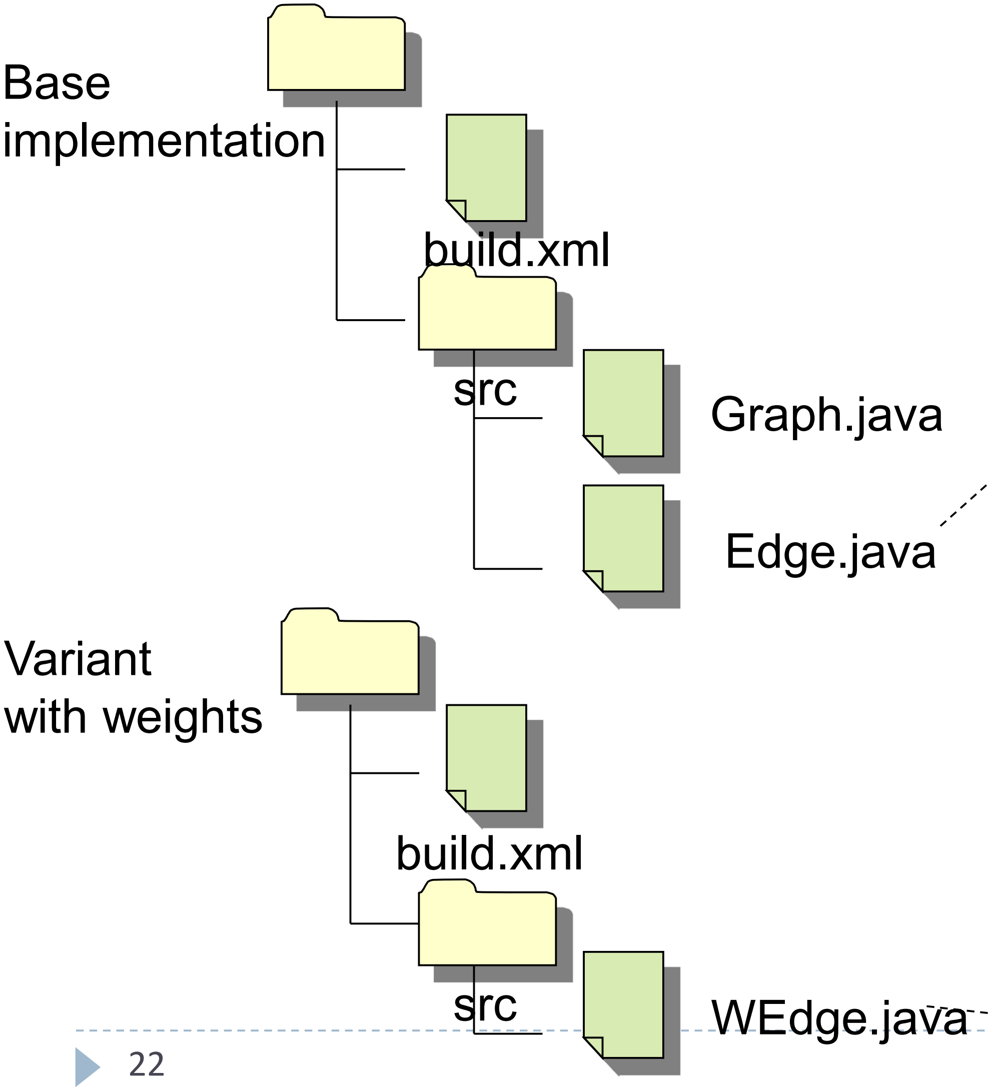
Example: graph library



```
class Edge {
  Node a, b;
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    a.print(); b.print();
  }
}
```

```
class Edge {
  Node a, b;
  Weight weight;
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    a.print(); b.print();
    weight.print();
  }
}
```

Alternative example: graph library



Base implementation

build.xml

src

Graph.java

Edge.java

Variant with weights

build.xml

src

WEdge.java

```
class Edge {
  Node a, b;
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    a.print(); b.print();
  }
}
```

```
class WEdge extends Edge {
  Weight weight;
  void print() {
    super.print();
    weight.print();
  }
}
```

…and adapt all calls of constructor;
might make sense to apply factory pattern

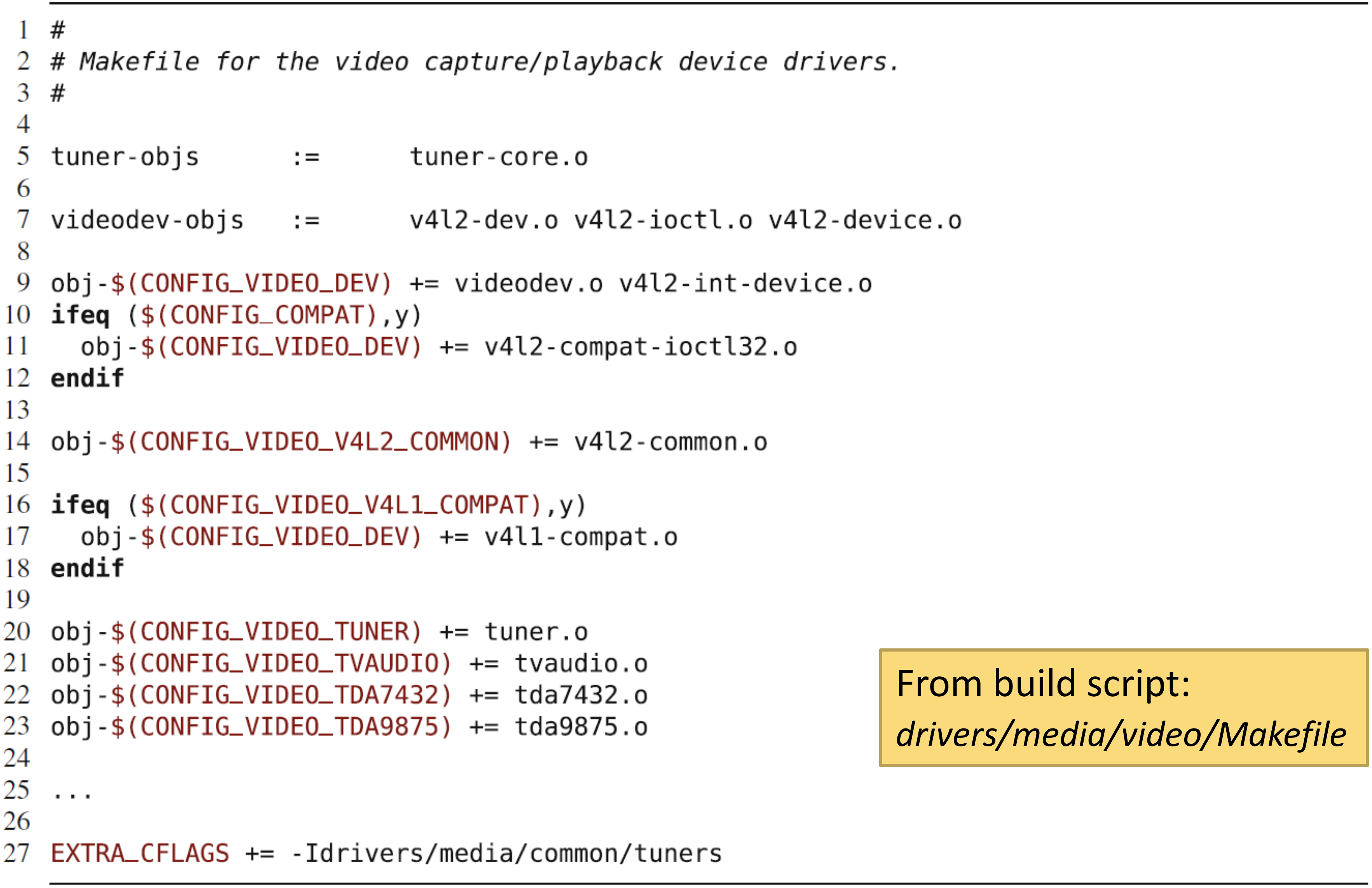
Solution 3: variability-based build system

- Example: Linux kernel
- Linux kernel has its own build system: kbuild
    - Maintains a hierarchy of >600 build scripts as input for make tool; structured by convention
    - Many scripts only executed if a particular option active
    - 97% of >9000 C files of linux kernel are optional

Solution 3: variability-based build system

- Typical commands
  - obj-y += foo.o Compile **foo.c** and link it to kernel
  - obj-m += foo.o Compile **foo.c** as a loadable kernel module
  - obj-l += foo.o Compile **foo.c** as a library
  - obj-(CONFIG_FOO) += foo.o Proceed with **foo.c** as specified in configuration: y, m, or n (don‘t compile)
- Additional patterns
  - Group files that belong to the same feature
  - Can set config options for all vs. selected files
  - Complex conditions: Kbuild contains a Turing-complete language

Build script from linux kernel



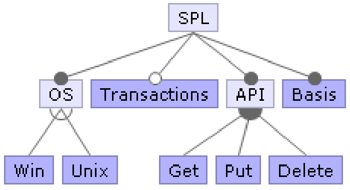
```
#
# Makefile for the video capture/playback device drivers.
#

tuner-objs      :=      tuner-core.o

videodev-objs   :=      v4l2-dev.o v4l2-ioctl.o v4l2-device.o

obj-$(CONFIG_VIDEO_DEV) += videodev.o v4l2-int-device.o
ifeq ($(CONFIG_COMPAT),y)
  obj-$(CONFIG_VIDEO_DEV) += v4l2-compat-ioctl32.o
endif

obj-$(CONFIG_VIDEO_V4L2_COMMON) += v4l2-common.o

ifeq ($(CONFIG_VIDEO_V4L1_COMPAT),y)
  obj-$(CONFIG_VIDEO_DEV) += v4l1-compat.o
endif

obj-$(CONFIG_VIDEO_TUNER) += tuner.o
obj-$(CONFIG_VIDEO_TVAUDIO) += tvaudio.o
obj-$(CONFIG_VIDEO_TDA7432) += tda7432.o
obj-$(CONFIG_VIDEO_TDA9875) += tda9875.o

...

EXTRA_CFLAGS += -Idrivers/media/common/tuners
```

From build script:
drivers/media/video/Makefile

Product lines with build systems

Domain Eng.

Feature model

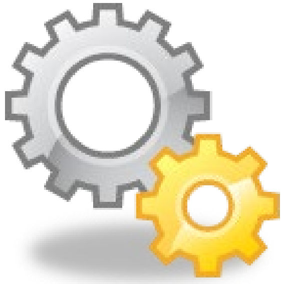


Base implementation

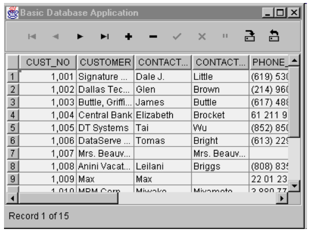
Application Eng.

- ☐Sensor-DB (Car)
- ☑Sensor-DB (Habitat Monitoring)
- ☐Sensor-DB (Earth quake)
- ☐SmartCard-DB
- ☐GPS-DB

Build script per variant
+ specific files



Standard build
(make, ant, …)



Basic Database Application

	CUST_NO	CUSTOMER	CONTACT...	CONTACT...	PHONE_
1	1,001	Signature ...	Dale J.	Little	(619) 530
2	1,002	Dallas Tec...	Glen	Brown	(214) 960
3	1,003	Buttle, Griffi...	James	Buttle	(617) 488
4	1,004	Central Bank	Elizabeth	Brocket	61 211 9
5	1,005	DT Systems	Tai	Wu	(852) 850
6	1,006	DataServe ...	Tomas	Bright	(613) 229
7	1,007	Mrs. Beauv...		Mrs. Beauv...	
8	1,008	Anini Vacat...	Leilani	Briggs	(808) 835
9	1,009	Max	Max		22 01 23
	1,010	MBM Corp...	Miwako	Miyamoto	3 880 77

Record 1 of 15

Complete program

Product lines with build systems – discussion

- Benefits
  - Relatively simple
  - Very flexible – arbitrary changes per variant
  - Little preparation/preplanning required
- Drawbacks
  - Each variant developed seperately; higher application engineering effort
  - Changes only at file granularity (can only override whole files)
  - Changing the base variant has many hard-to-foresee consequences

Outlook

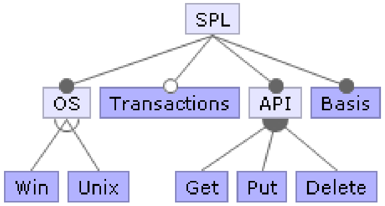
- More compile-time variability mechanisms
  - more fine-grained changes
  - develop features instead of variants

Preprocessors

How to implement variability?

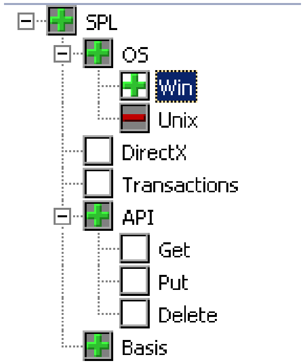
Domain Eng.

Feature model



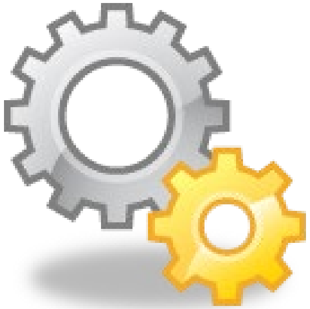
Reusable implementation artifacts

Application Eng.

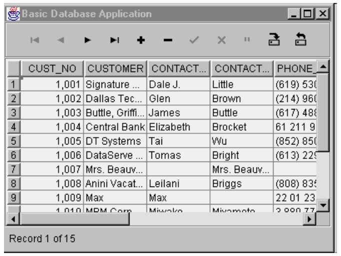


- ⊟ ☑ SPL
  - ⊟ ☑ OS
    - ☑ Win
    - ☒ Unix
  - □ DirectX
  - □ Transactions
  - ⊟ ☑ API
    - □ Get
    - □ Put
    - □ Delete
  - ☑ Basis

Feature selection



Generator



	CUST_NO	CUSTOMER	CONTACT...	CONTACT...	PHONE_
1	1,001	Signature ...	Dale J.	Little	(619) 53
2	1,002	Dallas Tec...	Glen	Brown	(214) 96
3	1,003	Buttle, Griffi...	James	Buttle	(617) 48
4	1,004	Central Bank	Elizabeth	Brocket	61 211 9
5	1,005	DT Systems	Tai	Wu	(852) 85
6	1,006	DataServe ...	Tomas	Bright	(613) 22
7	1,007	Mrs. Beauv...		Mrs. Beauv...	
8	1,008	Anini Vacat...	Leilani	Briggs	(808) 83
9	1,009	Max	Max		22 01 23
	1,010	MBM Corp	Miwako	Miyamoto	3 880 77

Record 1 of 15

Final program

Variability at compile time

- Goal: only compile code required in product
- But features freely selectable

What‘s missing?

- „if“, but evaluated already at compile time
- remove entire methods and classes if desired
- allow alternative implementations

```
class Conf {
   public static boolean COLORED = true;
   public static boolean WEIGHTED = false;
}
```

truly remove

remove

```
class Graph {
  Vector nv = new Vector(); Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    if (Conf.WEIGHTED) e.weight = new Weight();
    return e;
  }
  Edge add(Node n, Node m, Weight w)
    if (!Conf.WEIGHTED) throw RuntimeException();
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = w; return e;
  }
  void print() {
    for(int i = 0; i < ev.size(); i++) {
      ((Edge)ev.get(i)).print();
    }
  }
}
```

remove

```
class Color {
  static void setDisplayColor(Color c) { ... }
}
```

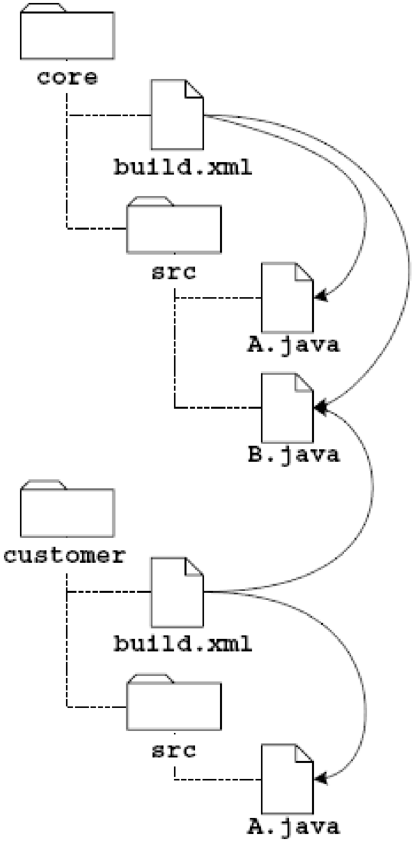
```
class Node {
  int id = 0;
  Color color = new Color();
  void print() {
   if (Conf.COLORED) Color.setDisplayColor(color);
    System.out.print(id);
  }
}
```

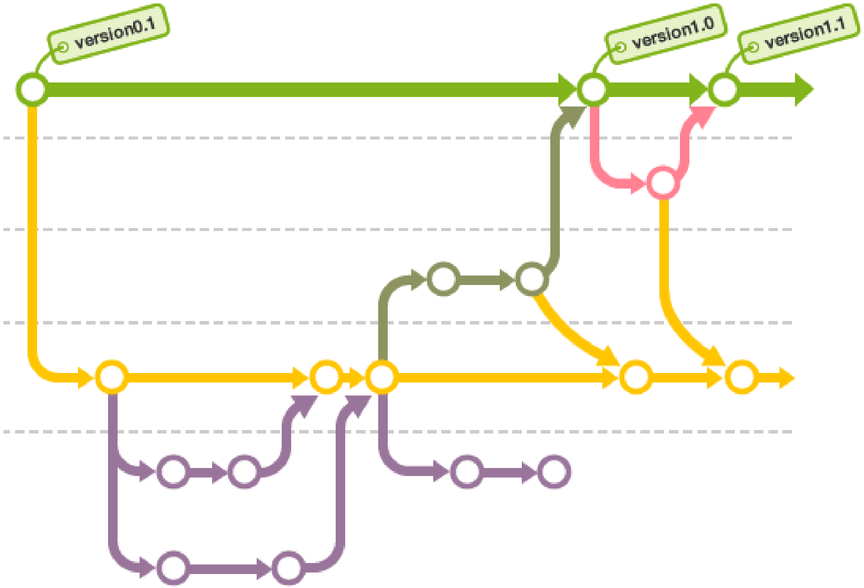
```
class Edge {
  Node a, b;
  Color color = new Color();
  Weight weight;
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    if (Conf. COLORED) Color.setDisplayColor(color);
    a.print(); b.print();
    if (!Conf.WEIGHTED)  weight.print();
  }
}
```

```
class Weight { void print() { ... } }
```

What’s missing?

- Feature-based planning and variant generation
- Fine-grained variant generation
- Make features explicit in source code





Preprocessors

- Transform the source code before the compiler is executed
- Expressiveness: from simple #include commands and boolean options to complex macro languages and rules
- Common in many programming languages
  - C, C++, Fortran, Erlang have a dedicated preprocessor
  - C#, Visual Basic, D, PL/SQL, Adobe Flex

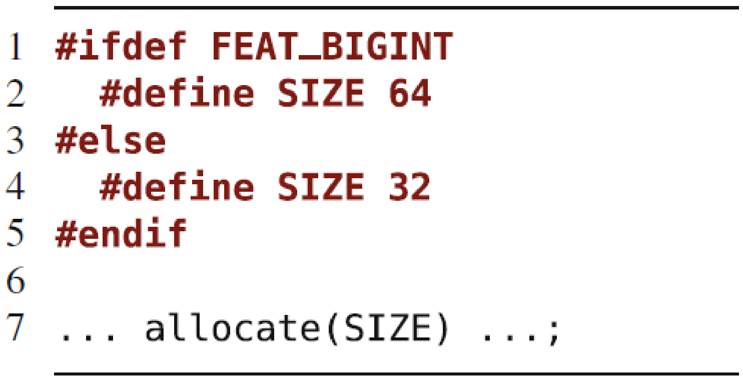
The C preprocessor *cpp:* Features

- **File includes**: #include
  - preprocessor inlines the to-be-included files
  - used, for example, for header files
- **Macros**: #define, #redefine
  - preprocessor replaces every occurrence of the given token by the given token sequence
  - used for defining compile-time functions and variables
- **Conditional compilation**: #if, #ifdef, #ifndef, #else
  - preprocessor removes token sequence before compilation
  - #if supports evaluation of expressions
  - #ifdef, #ifndef: check if feature set or unset

#ifdef example from BerkeleyDB

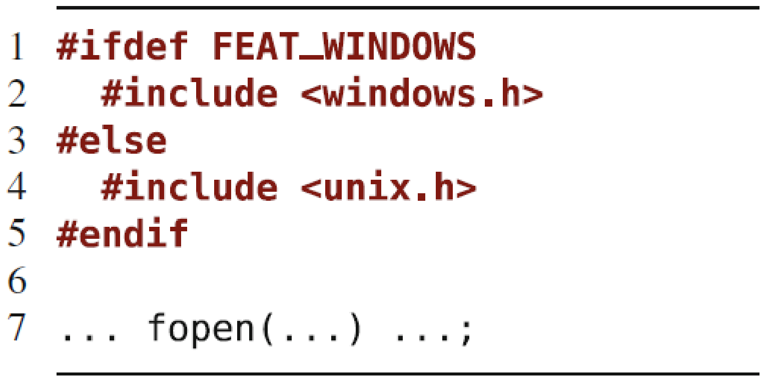
```
static int __rep_queue_filedone(dbenv, rep, rfp)
        DB_ENV *dbenv;
        REP *rep;
        __rep_fileinfo_args *rfp; {
#ifndef HAVE_QUEUE
        COMPQUIET(rep, NULL);
        COMPQUIET(rfp, NULL);
        return (__db_no_queue_am(dbenv));
#else
        db_pgno_t first, last;
        u_int32_t flags;
        int empty, ret, t_ret;
#ifdef DIAGNOSTIC
        DB_MSGBUF mb;
#endif
        // over 100 lines of additional code
}
#endif
```

Variability with cpp: typical patterns



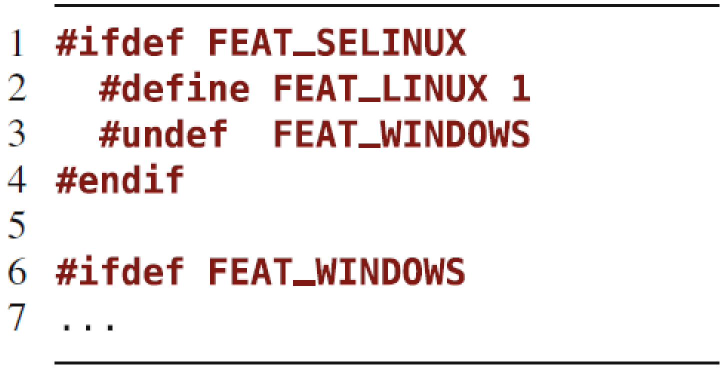
```
#ifdef FEAT_BIGINT
  #define SIZE 64
#else
  #define SIZE 32
#endif

... allocate(SIZE) ...;
```

Alternative macro definitions



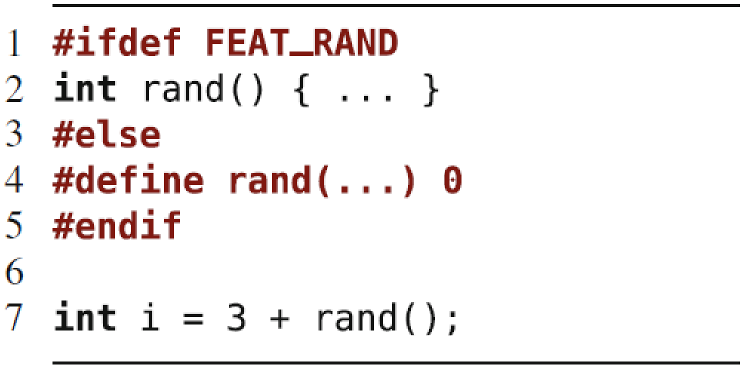
```
#ifdef FEAT_WINDOWS
  #include <windows.h>
#else
  #include <unix.h>
#endif

... fopen(...) ...;
```

Alternative Includes



```
#ifdef FEAT_SELINUX
  #define FEAT_LINUX 1
  #undef  FEAT_WINDOWS
#endif

#ifdef FEAT_WINDOWS
...
```

Conditional feature definitions



```
#ifdef FEAT_RAND
int rand() { ... }
#else
#define rand(...) 0
#endif

int i = 3 + rand();
```

Alternative function definitions

Preprocessor in Java?

- Not natively available
- Conditional compilation possible in some cases (as a compiler optimisation); only statement level

```
class Example {
    public static final boolean DEBUG = false;

    void main() {
        System.out.println(“always”);
        if (DEBUG)
            System.out.println(“debug info”);
    }
}
```

- External tools, e.g. CPP, Antenna, Munge, XVCL, Gears, pure::variants

Munge

- Simple preprocessor for Java
- Originally for Swing in Java 1.2

```
class Example {
void main() {
		System.out.println(“immer”);
		/*if[DEBUG]*/
		System.out.println(“debug info”);
		/*end[DEBUG]*/
}
}
```

java Munge –DDEBUG –DFEATURE2 File1.java File2.java ... target-directory

feature selection according to feature model

http://weblogs.java.net/blog/tball/archive/2006/09/munge_swings_se.html

Refresher: graph example

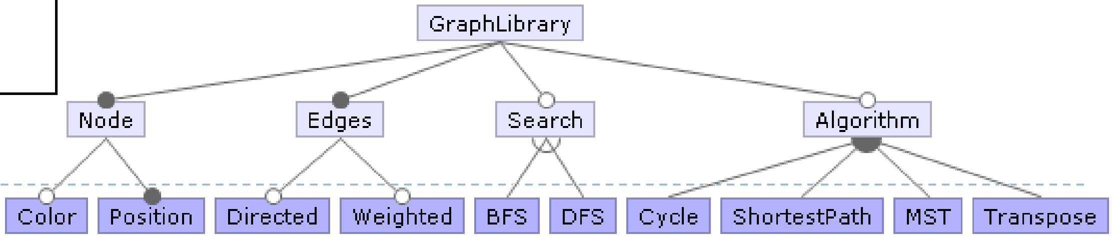
```
class Graph {
  Vector nv = new Vector(); Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = new Weight();
    return e;
  }
  Edge add(Node n, Node m, Weight w)
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = w; return e;
  }
  void print() {
    for(int i = 0; i < ev.size(); i++) {
      ((Edge)ev.get(i)).print();
    }
  }
}
```

```
class Color {
  static void setDisplayColor(Color c) { ... }
}
```

```
class Weight { void print() { ... } }
```

```
class Node {
  int id = 0;
  Color color = new Color();
  void print() {
    Color.setDisplayColor(color);
    System.out.print(id);
  }
}
```

```
class Edge {
  Node a, b;
  Color color = new Color();
  Weight weight;= new Weight();
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    Color.setDisplayColor(color);
    a.print(); b.print();
    weight.print();
  }
}
```



Graph example with Munge

```
class Graph {
  Vector nv = new Vector(); Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    /*if[WEIGHT]*/
    e.weight = new Weight();
    /*end[WEIGHT]*/
    return e;
  }
  /*if[WEIGHT]*/
  Edge add(Node n, Node m, Weight w)
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = w; return e;
  }
  /*end[WEIGHT]*/
  void print() {
    for(int i = 0; i < ev.size(); i++) {
      ((Edge)ev.get(i)).print();
    }
  }
}
```

```
/*if[WEIGHT]*/
class Weight { void print() { ... } }
/*end[WEIGHT]*/
```

```
class Edge {
  Node a, b;
  /*if[COLOR]*/
  Color color = new Color();
  /*end[COLOR]*/
  /*if[WEIGHT]*/
  Weight weight;
  /*end[WEIGHT]*/
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    /*if[COLOR]*/
    Color.setDisplayColor(color);
    /*end[COLOR]*/
    a.print(); b.print();
    /*if[WEIGHT]*/
    weight.print();
    /*end[WEIGHT]*/
  }
}
```

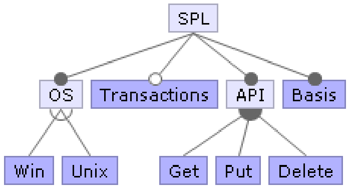
```
/*if[COLOR]*/
class Color {
  static void setDisplayColor(Color c) { ... }
}
/*end[COLOR]*/
```

```
class Node {
  int id = 0;
  /*if[COLOR]*/
```

Product lines with preprocessor

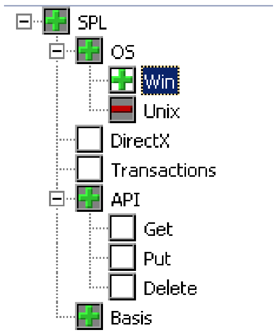
Domain Eng.

feature model



program with preprocessor directives

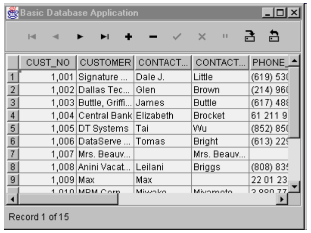
Application Eng.



- ⊟ ☑ SPL
  - ⊟ ☑ OS
    - ☑ Win
    - ☒ Unix
  - ☐ DirectX
  - ☐ Transactions
  - ⊟ ☑ API
    - ☐ Get
    - ☐ Put
    - ☐ Delete
  - ☑ Basis

feature selection

preprocessor



Basic Database Application

	CUST_NO	CUSTOMER	CONTACT...	CONTACT...	PHONE_
1	1,001	Signature ...	Dale J.	Little	(619) 530
2	1,002	Dallas Tec...	Glen	Brown	(214) 960
3	1,003	Buttle, Griffi...	James	Buttle	(617) 488
4	1,004	Central Bank	Elizabeth	Brocket	61 211 9
5	1,005	DT Systems	Tai	Wu	(852) 850
6	1,006	DataServe ...	Tomas	Bright	(613) 229
7	1,007	Mrs. Beauv...		Mrs. Beauv...	
8	1,008	Anini Vacat...	Leilani	Briggs	(808) 835
9	1,009	Max	Max		22 01 23
	1,010	MRM Corp...	Miwako	Miyamoto	3 880 77

Record 1 of 15

complete program

Further preprocessors

XVCL

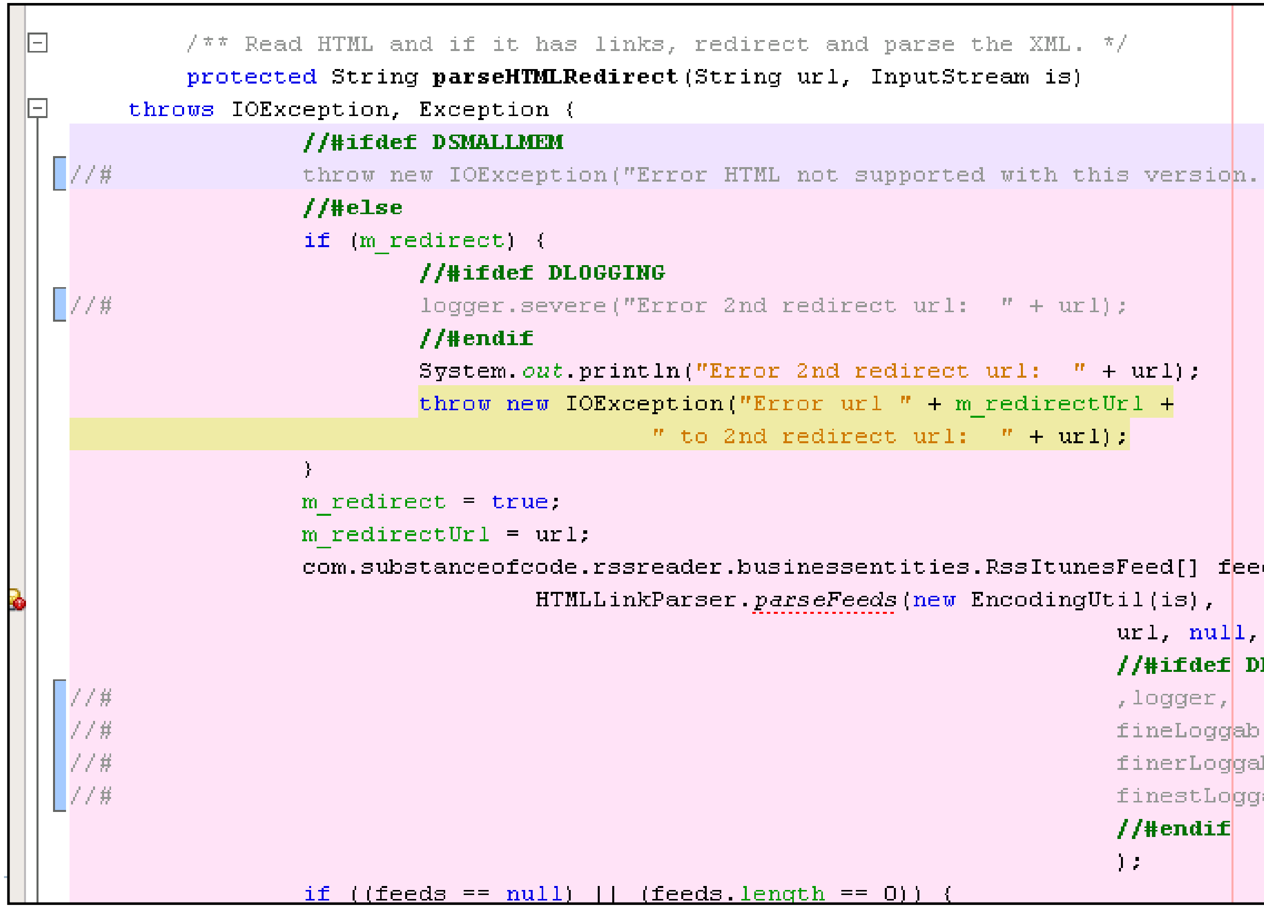
- XML-based preprocessor
- Based on a hierarchy of frames

```
<x-frame name="Notepad">
import java.awt.*;
class Notepad extends JPanel {
    Notepad() {
        super();
        …
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.setTitle("<value-of expr="?@TITLE?"/>");
        frame.setBackground(
            Color.<value-of expr="?@BGCOLOR?"/>);
        frame.show();
    }
    <adapt x-frame="Editor.XVCL"/>
    <adapt x-frame="Menubar.XVCL"/>
    <adapt x-frame="Toolbar.XVCL"/>
    …
}
</x-frame>
```

```
<x-frame name="Toolbar">
<set-multivar="ToolbarBtns" value="New,Open,Save"/>
private Component createToolbar() {
      JToolBar toolbar = new JToolBar();
      JButton button;
<while using-items-in="ToolbarBtns">
   <select option="ToolbarBtns">
      <option value="-">
      toolbar.add(Box.createHorizontalStrut(5));
      </option>
      <otherwise>
      button = new JButton(new ImageIcon(
         "<value-of expr="?@Gif@ToolbarBtns?"/> "));
      toolbar.add(button);
      </otherwise>
   </select>
</while>
      toolbar.add(Box.createHorizontalGlue());
      return toolbar;
}
</x-frame>
```

Antenna

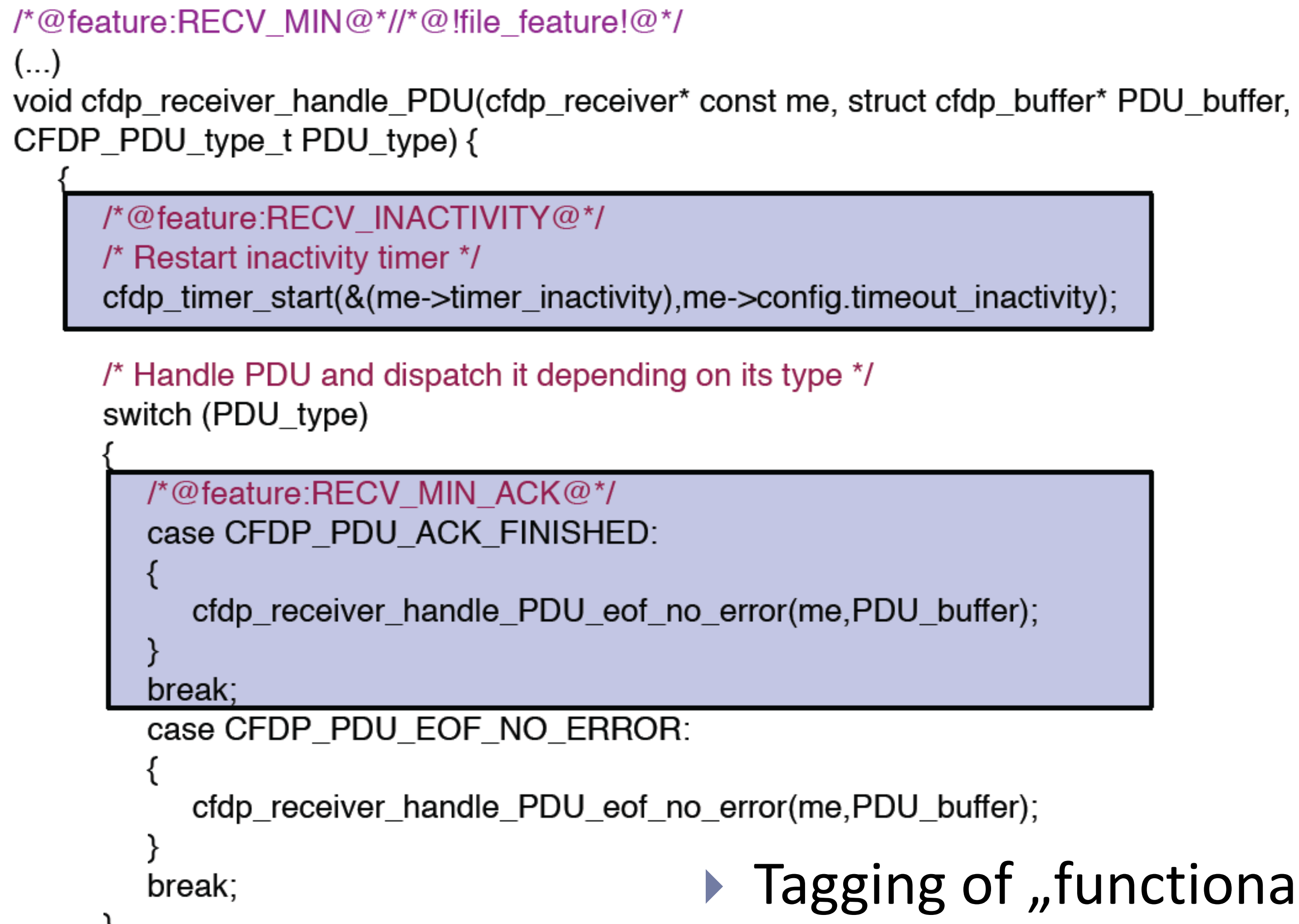
- Collection of ant tasks for Java ME
  - Java ME: early Java implementation for embedded systems
- Contains #ifdef directive, like *cpp*
- Used in many Java ME projects



```
        /** Read HTML and if it has links, redirect and parse the XML. */
        protected String parseHTMLRedirect(String url, InputStream is)
    throws IOException, Exception {
                //#ifdef DSMALLMEM
//#             throw new IOException("Error HTML not supported with this version.
                //#else
                if (m_redirect) {
                        //#ifdef DLOGGING
//#                     logger.severe("Error 2nd redirect url:  " + url);
                        //#endif
                        System.out.println("Error 2nd redirect url:  " + url);
                        throw new IOException("Error url " + m_redirectUrl +
                                        " to 2nd redirect url:  " + url);
                }
                m_redirect = true;
                m_redirectUrl = url;
                com.substanceofcode.rssreader.businessentities.RssItunesFeed[] fee
                                HTMLLinkParser.parseFeeds(new EncodingUtil(is),
                                                                        url, null,
                                                                        //#ifdef D
//#                                                                     ,logger,
//#                                                                     fineLoggab
//#                                                                     finerLogga
//#                                                                     finestLogg
                                                                        //#endif
                                                                        );
                if ((feeds == null) || (feeds.length == 0)) {
```

http://antenna.sourceforge.net/

Semantic preprocessors: tag and prune



```
/*@feature:RECV_MIN@*//*@!file_feature!@*/
(...)
void cfdp_receiver_handle_PDU(cfdp_receiver* const me, struct cfdp_buffer* PDU_buffer,
CFDP_PDU_type_t PDU_type) {
    {
        /*@feature:RECV_INACTIVITY@*/
        /* Restart inactivity timer */
        cfdp_timer_start(&(me->timer_inactivity),me->config.timeout_inactivity);

        /* Handle PDU and dispatch it depending on its type */
        switch (PDU_type)
        {
            /*@feature:RECV_MIN_ACK@*/
            case CFDP_PDU_ACK_FINISHED:
            {
                cfdp_receiver_handle_PDU_eof_no_error(me,PDU_buffer);
            }
            break;
            case CFDP_PDU_EOF_NO_ERROR:
            {
                cfdp_receiver_handle_PDU_eof_no_error(me,PDU_buffer);
            }
            break;
        }
        (...)
    }
}
```

- Tagging of „functional blocks“

Patrick Heymans, Quentin Boucher, Andreas Classen, Arnaud Bourdoux, Laurent Demonceau: A code tagging approach to software product line development - An application to satellite communication libraries. STTT 14(5): 553-566 (2012)

Discussion

Benefits

- Widely used
  - Contained in many language, tool support available
  - Familiar to developers who know these languages
- Simple
  - Very simple programming concept:
    mark and remove
  - Very flexible, expressive, fine-grained
- No preparation/preplanning necessary
  - Easy to introduce into existing project

Main problem: code readability

- Mixes two languages (C and #ifdefs, or Java and Munge, …)
- Understanding of control flow complicated
- Long annotated code blocks: beginning and end hard to find
- Additional line breaks corrupt layout

```
class Stack {
void push(Object o
#ifdef SYNC
, Transaction txn
#endif
) {
    if (o==null
#ifdef SYNC
        || txn==null
#endif
        ) return;
#ifdef SYNC
    Lock l=txn.lock(o);
#endif
    elementData[size++] = o;
#ifdef SYNC
    l.unlock();
#endif
    fireStackChanged();
}}
```

➔ Modularity as alternative?

Further problems

- Complexity due to arbitrarily deep nesting
- Especially error-prone when used in complex code constructs, uncontrolled/undisciplined use
- Examples:
  - Variable return type

```
/*if[WEIGHT]*/W/*end[WEIGHT]*/Edge add(Node n, Node m /*if[WEIGHT]*/, int w/*end[WEIGHT]*/) {
   return new /*if[WEIGHT]*/W/*end[WEIGHT]*/Edge (n, m /*if[WEIGHT]*/, w/*end[WEIGHT]*/ );
}
```

  - Commas between multiple parameters

```
Edge set(/*if[WEIGHT]*/int w/*if[COLOR]*/, /*end[COLOR]*/ /*end[WEIGHT]*/ /*if[COLOR]*/int c/*end[COLOR]*/ ) {
   …
}
```

Problem: error-prone

- Syntax errors
- Type errors

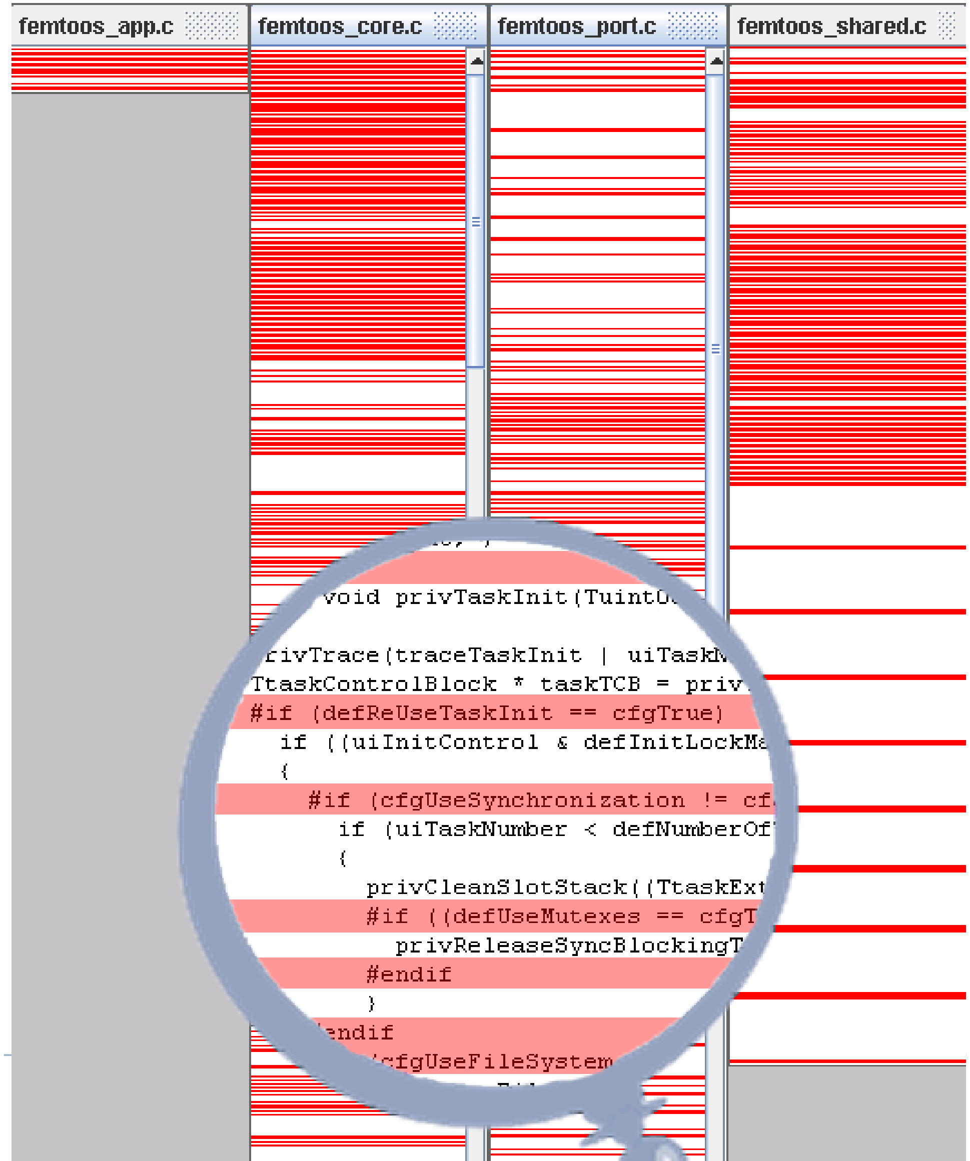
```
static int _rep_queue_filedone(...)
    DB_ENV *dbenv;
    REP *rep;
    __rep_fileinfo_args *rfp; {
#ifndef HAVE_QUEUE
    COMPQUIET(rep, NULL);
    COMPQUIET(rfp, NULL);
    return (__db_no_queue_am(dbenv));
#else
    db_pgno_t first, last;
    u_int32_t flags;
    int empty, ret, t_ret;
#ifdef DIAGNOSTIC
    DB_MSGBUF mb;
#endif
//over 100 lines of additional code
}
#endif
```

```
#ifdef TABLES
class Table {
  void insert(Object data,
        Txn txn) {
       storage.set(data,
        txn.getLock());
   }
}
#endif
class Storage {
#ifdef WRITE
   boolean set(…) { ... }
#endif
}
```

Additional problems

- Feature core is spread throughout entire program
  - ➔ *feature traceability problem*
  - How to find a bug in feature *Weight*?
- Tool support becomes much more complicated
  - Experience from C/C++ (refactoring, analysis, …)
  - Munge and others: definitions in comments

Preprocessor in Femto OS



```
  void privTaskInit(TuintO
 
rivTrace(traceTaskInit | uiTaskN
TtaskControlBlock * taskTCB = priv
#if (defReUseTaskInit == cfgTrue)
  if ((uiInitControl & defInitLockMa
  {
    #if (cfgUseSynchronization != cf
      if (uiTaskNumber < defNumberOf
      {
        privCleanSlotStack((TtaskExt
        #if ((defUseMutexes == cfgT
          privReleaseSyncBlockingT
        #endif
        }
endif
   /cfgUseFileSystem
```

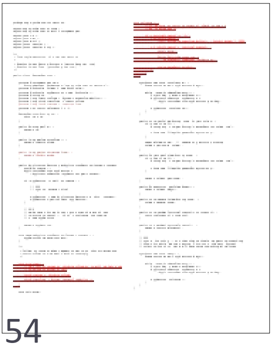
A question of size

- Example: Session expiration in the Apache Tomcat Server

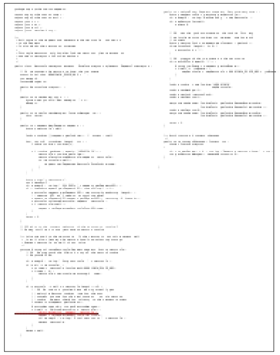
ApplicationSession

StandardSession

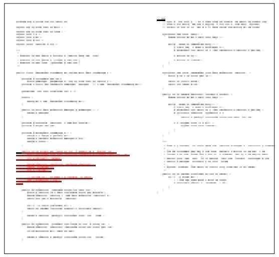
ServerSession



SessionInterceptor

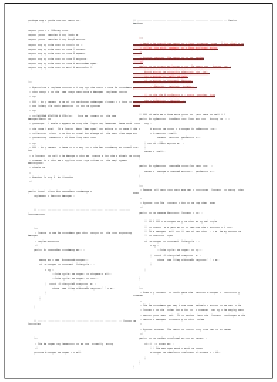


ServerSessionManager



StandardManager

StandardSessionManager

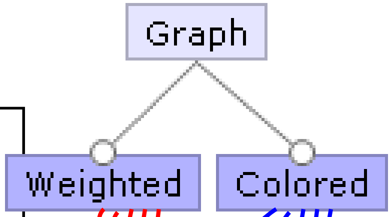


Feature traceability problem

Problem: implementation scattered

- Features vanish in implementation
  - What belongs to a feature?
  - Maintaining the feature might require to find all relevant code parts
- Collaboration is made more complicated
  - Experts for different features might have to work on same code files at the same time
- Roadblock to productivity and efficient evolution
  - When adding new functionality, developer has to think about other concerns that are not directly relevant for the task at hand (readability, understandability)

Features in graph example



```
class Graph {
  Vector nv = new Vector(); Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = new Weight();
    return e;
  }
  Edge add(Node n, Node m, Weight w)
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = w; return e;
  }
  void print() {
    for(int i = 0; i < ev.size(); i++) {
      ((Edge)ev.get(i)).print();
    }
  }
}
```

```
class Node {
  int id = 0;
  Color color = new Color();
  void print() {
    Color.setDisplayColor(color);
    System.out.print(id);
  }
}
```

```
class Edge {
  Node a, b;
  Color color = new Color();
  Weight weight;= new Weight();
  Edge(Node _a, Node _b) { a = _a; b = _b; }
  void print() {
    Color.setDisplayColor(color);
    a.print(); b.print();
    weight.print();
  }
}
```

```
class Color {
  static void setDisplayColor(Color c) { ... }
}
```

```
class Weight { void print() { ... } }
```

Consequences

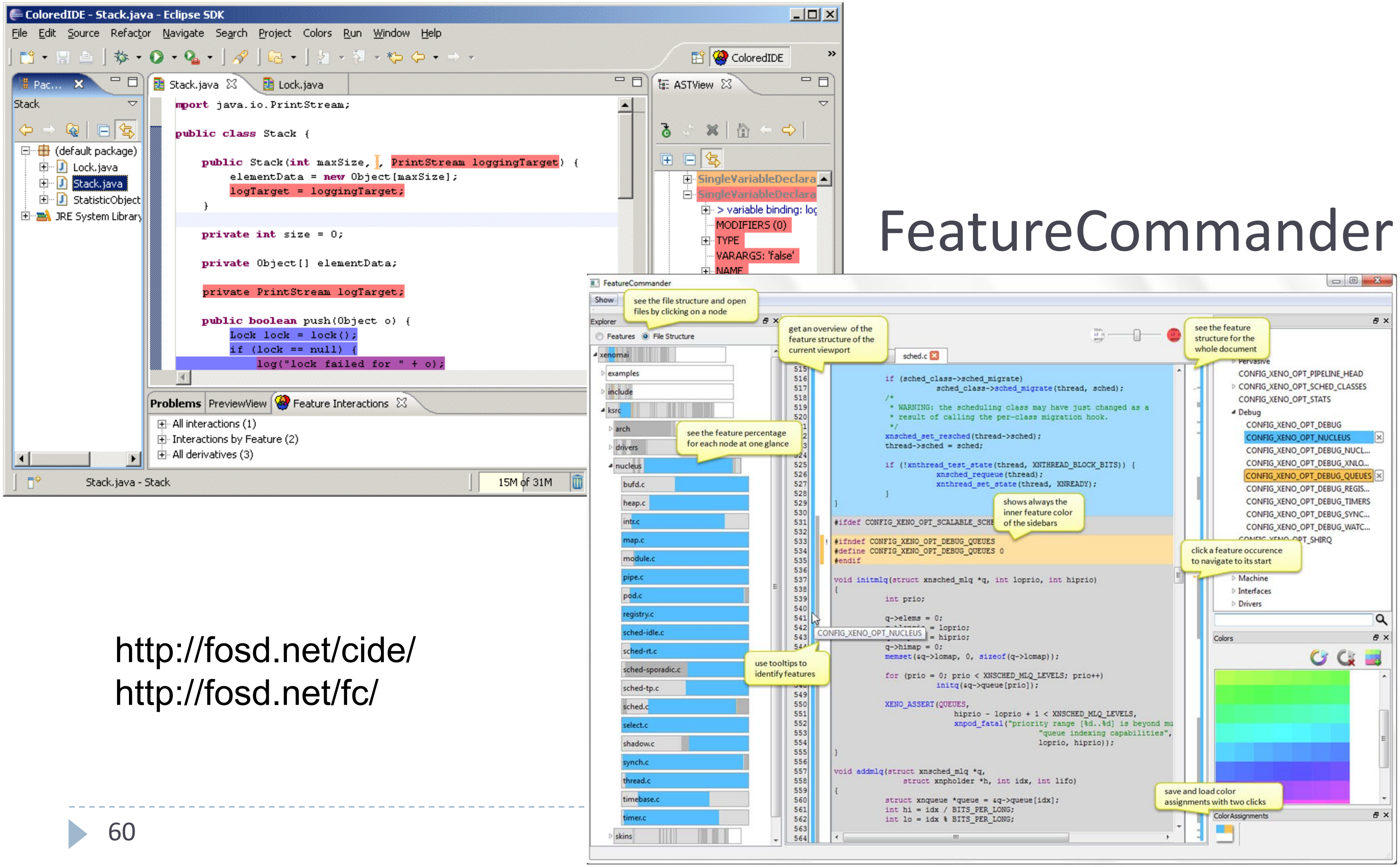
```
class BusinessClass
  //... data fields
  //... logging stream
  //... cache status
  public void importantOperation(
        Data data, User currentUser, ...){
    // check authorization
    // lock objects for synchronization
    // check if puffer up-to-date
    // log start of actual operation
    // execute actual operation
    // log end of actual operation
    // unlock objects
  }
  public void alsoImportantOperation(
        OtherData data, User currentUser, ...){
    // check authorization
    // lock objects for synchronization
    // check if puffer up-to-date
    // log start of actual operation
    // execute actual operation
    // log end of actual operation
    // unlock objects
  }
}
```

- Which code belongs to authentication?
- When locking procedure is to be changed: which parts have to be touched?
- A user could delete files without being logged in: where to search for error?

Feature traceability

- Want to keep feature-to-code connection
- Ideally, a module per feature
- Need to resort to workarounds and makeshift solutions if modularization not possible
  - Comments or annotations in source code (e.g., all code relevant for authentication marked with „//auth“)
  - Naming concentions (e.g., all authentication-related methods start with „auth_“)
  - Additional tools, e.g., as part of IDE
- Preprocessors already offer annotations
  - But usually only optional features

CIDE



```
mport java.io.PrintStream;

public class Stack {

    public Stack(int maxSize, |, PrintStream loggingTarget) {
        elementData = new Object[maxSize];
        logTarget = loggingTarget;
    }

    private int size = 0;

    private Object[] elementData;

    private PrintStream logTarget;

    public boolean push(Object o) {
        Lock lock = lock();
        if (lock == null) {
            log("lock failed for " + o);
```

- All interactions (1)
- Interactions by Feature (2)
- All derivatives (3)

FeatureCommander

see the file structure and open files by clicking on a node

get an overview of the feature structure of the current viewport

see the feature structure for the whole document

see the feature percentage for each node at one glance

shows always the inner feature color of the sidebars

click a feature occurence to navigate to its start

use tooltips to identify features

save and load color assignments with two clicks

```
			if (sched_class->sched_migrate)
				sched_class->sched_migrate(thread, sched);
			/*
			 * WARNING: the scheduling class may have just changed as a
			 * result of calling the per-class migration hook.
			 */
			xnsched_set_resched(thread->sched);
			thread->sched = sched;

			if (!xnthread_test_state(thread, XNTHREAD_BLOCK_BITS)) {
				xnsched_requeue(thread);
				xnthread_set_state(thread, XNREADY);
			}
}

#ifdef CONFIG_XENO_OPT_SCALABLE_SCHE

#ifndef CONFIG_XENO_OPT_DEBUG_QUEUES
#define CONFIG_XENO_OPT_DEBUG_QUEUES 0
#endif

void initmlq(struct xnsched_mlq *q, int loprio, int hiprio)
{
			int prio;

			q->elems = 0;
			q->loprio = loprio;
			q->hiprio = hiprio;
			q->himap = 0;
			memset(&q->lomap, 0, sizeof(q->lomap));

			for (prio = 0; prio < XNSCHED_MLQ_LEVELS; prio++)
				initq(&q->queue[prio]);

			XENO_ASSERT(QUEUES,
				hiprio - loprio + 1 < XNSCHED_MLQ_LEVELS,
				xnpod_fatal("priority range [%d..%d] is beyond mu
						"queue indexing capabilities",
						loprio, hiprio));
}

void addmlq(struct xnsched_mlq *q,
		struct xnpholder *h, int idx, int lifo)
{
			struct xnqueue *queue = &q->queue[idx];
			int hi = idx / BITS_PER_LONG;
			int lo = idx % BITS_PER_LONG;
```

http://fosd.net/cide/
http://fosd.net/fc/

Outlook

- Feature implementation in modular way
- Dealing with cross-cutting concerns
- Possible improvements of preprocessor

Literature

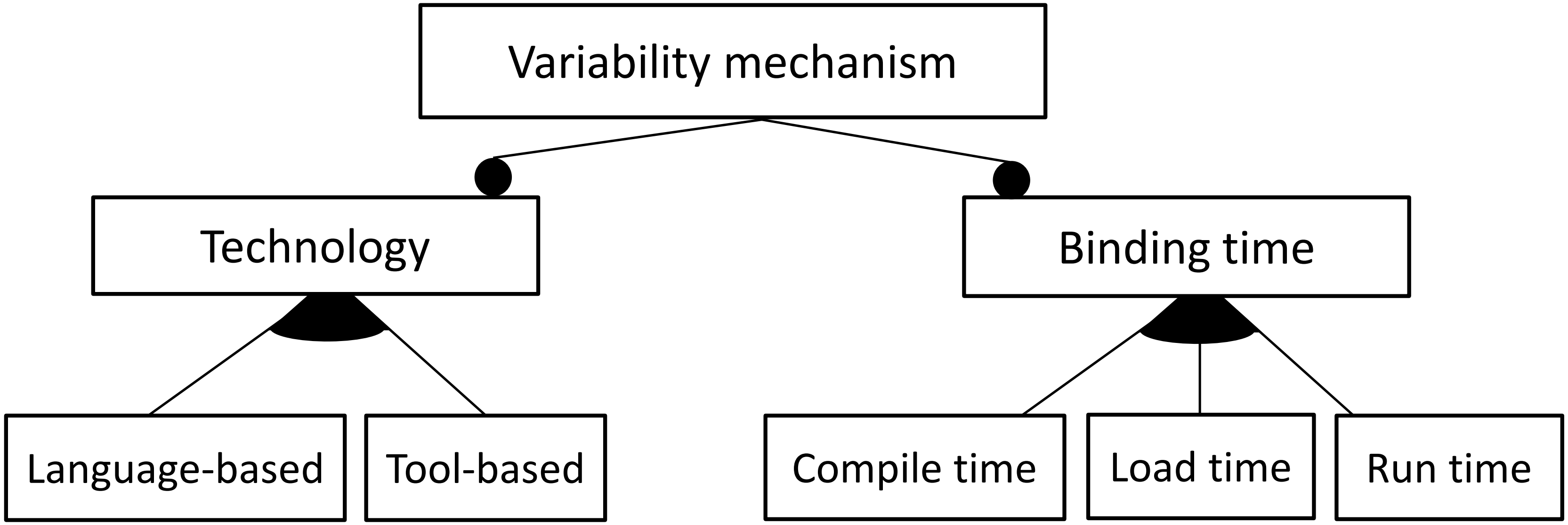
- M. Staples, D. Hill. Experiences adopting software product line development without a product line architecture. Proceedings APSEC, pp. 176—183, 2004

  [Industrial experience of using version control and build systems for product line development]

- T. Dhaliwal, F. Khomh, Y. Zou, A. Hassan. Recovering commit dependencies for selective code integration in software product lines. Proceedings ICSM, 202—211, 2012

  [Mapping of commits to features; dependency analysis]

Zoom quiz: versioning systems, build systems?



Zoom quiz

- How many source code variants are possible?

```
int a = 1;
#ifdef A
int c = a;
#endif
if (c) {
   c += a;
#ifdef A && B
   c /= a;
#endif
}
```

Zoom quiz

- Where is the bug?

(a)

```
int a = 1;
int b = 1;
#ifdef A
int c = a;
#else
char c = a;
#endif
if (c) {
  c += a;
#ifdef B
  c /= b;
}
#endif
```

(b)

```
int a = 1;
int b = 1;
#ifdef A
int c = a;
#else
char d = a;
#endif
if (c) {
  c += a;
#ifdef B
  c /= b;
#endif
}
```

(c)

```
int a = 1;
int b = 0;
#ifdef A
int c = a;
#else
char c = a;
#endif
if (c) {
  c += a;
#ifdef B
  c /= b;
#endif
}
```