Software Product Lines

15-214

Charlie Garrod, Christian Kästner

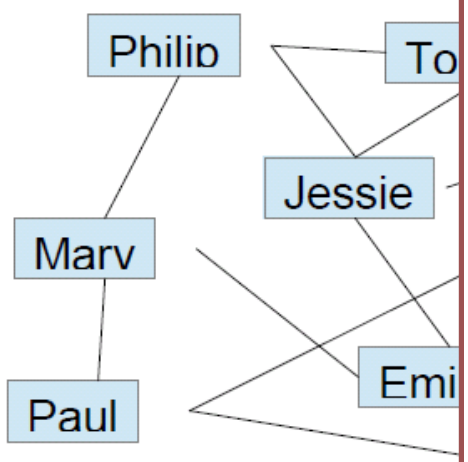
Reuse and Variations

15-214 Homework 0

Homework 0: A Friendship Graph

The goal of this homework is to get familiar with the infrastructure used in this course and to attempt some first steps in Java programming. To simplify the programming task, we have selected a simple task from a familiar domain.

Task: We want to represent the friendship
we want to calculate the shortest connectio
Test your implementation with the followin



We expect a client implementation roughly

```
FriendGraph g = new FriendGraph();
Person anne = new Person("Anne");
```

15-214 Homework 1

Homework #1: Graph Algorithms at Facecook

Due Thursday, January 31st at 11:59 p.m.

After graduating from CMU you have been hired by Facecook, an up-and-coming social network company, to develop analytical algorithms for their website. The Facecook design committee has developed a `Graph` interface that will be used to store social network information for Facecook users. You have decided to experiment with two different `Graph` implementations to find which representation is best for different purposes. Be careful! To ensure that your implementation works with existing Facecook code, you must not alter the `Graph` interface in any way!

The goals for this assignment are:

- Understand and apply the concepts of polymorphism and encapsulation
- Learn to use Javadoc
- Better understand graphs and their multiple representations
- Familiarize yourself with Java and Eclipse

Java - jeti/src/nu/fw/jeti/jabber/Backend.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | Java | Debug

Pac | Typ | JUni

- Composite
- Exam
- JBother
- JBother_
- > jeti 759 [https://jeti.svn.sourceforge
  - > src 759
    - > (default package) 759
    - buildfiles 742
    - com.swabunga.spell.engine 748
    - com.swabunga.spell.event 749
    - com.swabunga.util 751
    - docs 614
    - languages 702
    - nu.fw.jeti.applet 759
    - nu.fw.jeti.backend 759
    - nu.fw.jeti.backend.roster 719
    - nu.fw.jeti.events 706
    - nu.fw.jeti.images 718
    - nu.fw.jeti.jabber 759
      - Backend.java 759
      - JID.java 740
      - JIDStatus.java 413
      - UnknownJIDStatus.java 413
      - XDataCallback.java 114
      - XDataPanel.java 651
    - nu.fw.jeti.jabber.elements 759

Notifiers.java | Plugin.java | Backend.java

```
/**
 * @author E.S. de Boer
 */

public class Backend
{
    private Map<String,List<JETIListener>>  eventListeners;
    private Map<JID,JETIListener>  presenceListeners;
    private Map<JID,JETIListener>  messageListeners;
    private int identifier;
    private Connect connect;
    private Roster roster;
    private Server server;
    private Jeti main;
    private JFrame mainFrame;
    private Container ma
    private Start start;
    private IQTimerQueue
    private Handlers han
    //private Capabilitie

    public Backend(Start
    {
        eventListeners =
        iqTimerQueue = ne
```

Problems | @ Javadoc | Decla

```
<terminated> New_configuration [Jav
nu.fw.jeti.ui.LoginStatusWind
nu.fw.jeti.ui.login.LoginWind
nu.fw.jeti.ui.login.EditProfi
nu.fw.jeti.ui.LoginStatusWind
```

Task List

Find | All | Activ...

Connect Mylyn

Connect to your task and ALM tools or create a local task.

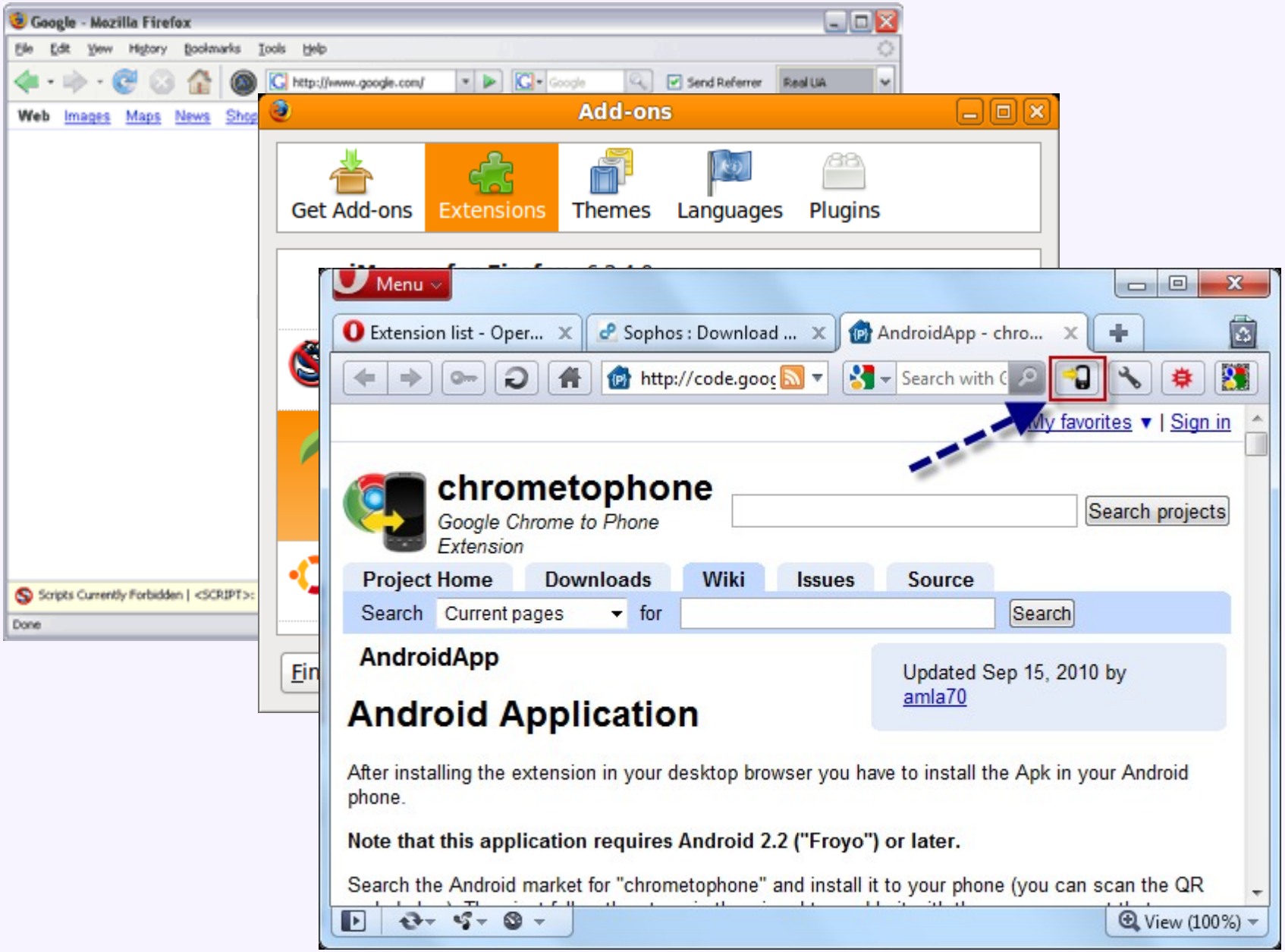
Outline

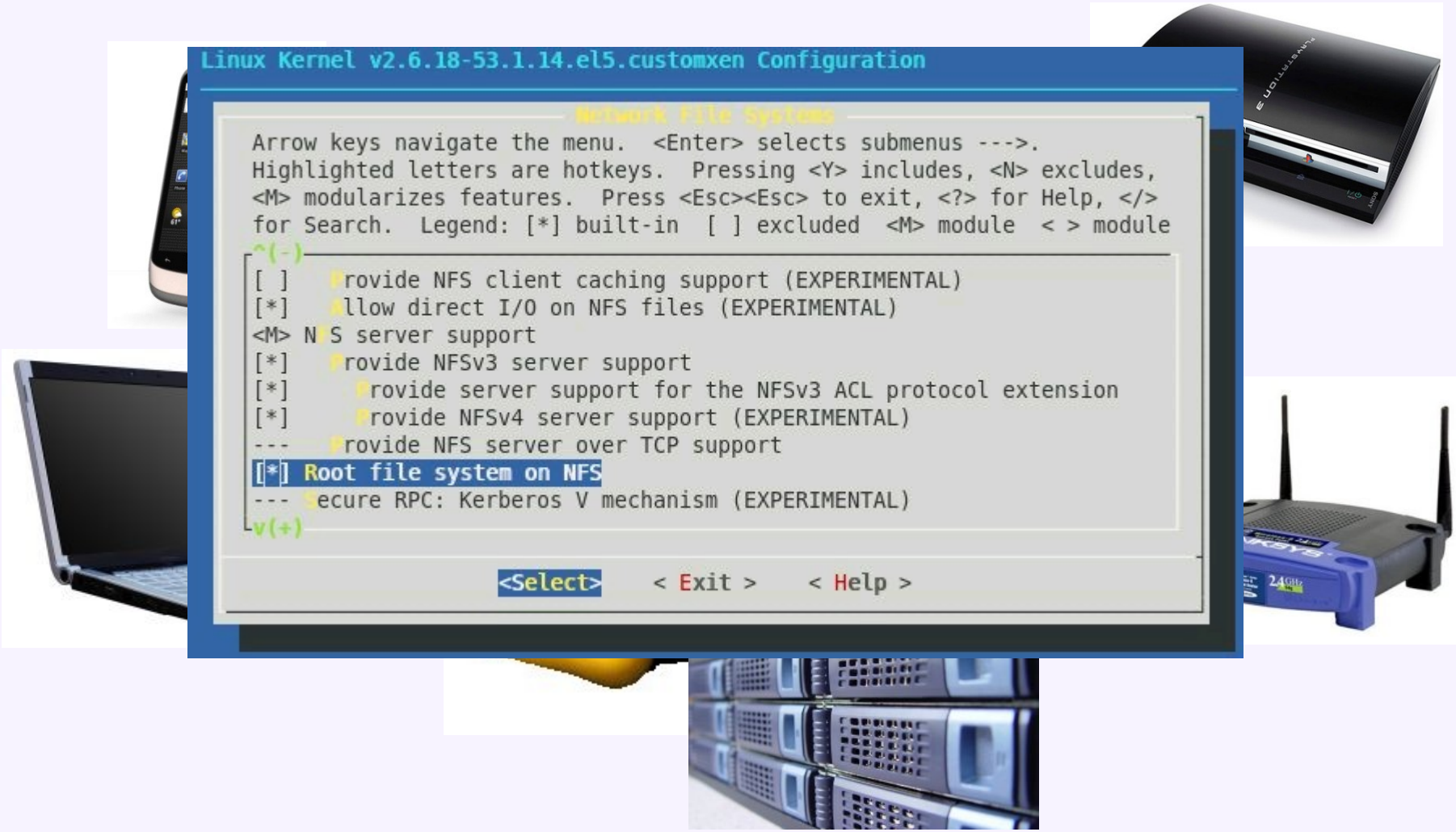
TuxGuitar - synth5.gp4

File Edit View Composition Track Measure Beat Marker Player Tools Help

N°	Name	Instrument
1	Piano	Piano
2	Solo	Jazz Electric Gtr
3	Bass	Fingered Bass

Right mode | Scale


Google - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.google.com/
Send Referrer
Web Images Maps News
Scripts Currently Forbidden
Done
Add-ons
Get Add-ons
Extensions
Themes
Languages
Plugins
Menu
Extension list - Oper...
Sophos : Download ...
AndroidApp - chro...
http://code.goog
Search with G
My favorites
Sign in
chrometophone
Google Chrome to Phone Extension
Search projects
Project Home
Downloads
Wiki
Issues
Source
Search
Current pages
for
Search
AndroidApp
Updated Sep 15, 2010 by amla70
Android Application
After installing the extension in your desktop browser you have to install the Apk in your Android phone.
Note that this application requires Android 2.2 ("Froyo") or later.
Search the Android market for "chrometophone" and install it to your phone (you can scan the QR
View (100%)


Linux Kernel v2.6.18-53.1.14.el5.customxen Configuration
Network File Systems
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < > module
[ ] Provide NFS client caching support (EXPERIMENTAL)
[*] Allow direct I/O on NFS files (EXPERIMENTAL)
<M> NFS server support
[*] Provide NFSv3 server support
[*] Provide server support for the NFSv3 ACL protocol extension
[*] Provide NFSv4 server support (EXPERIMENTAL)
--- Provide NFS server over TCP support
[*] Root file system on NFS
--- Secure RPC: Kerberos V mechanism (EXPERIMENTAL)
<Select> < Exit > < Help >

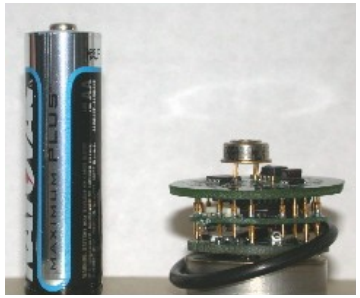
Reuse and Variation

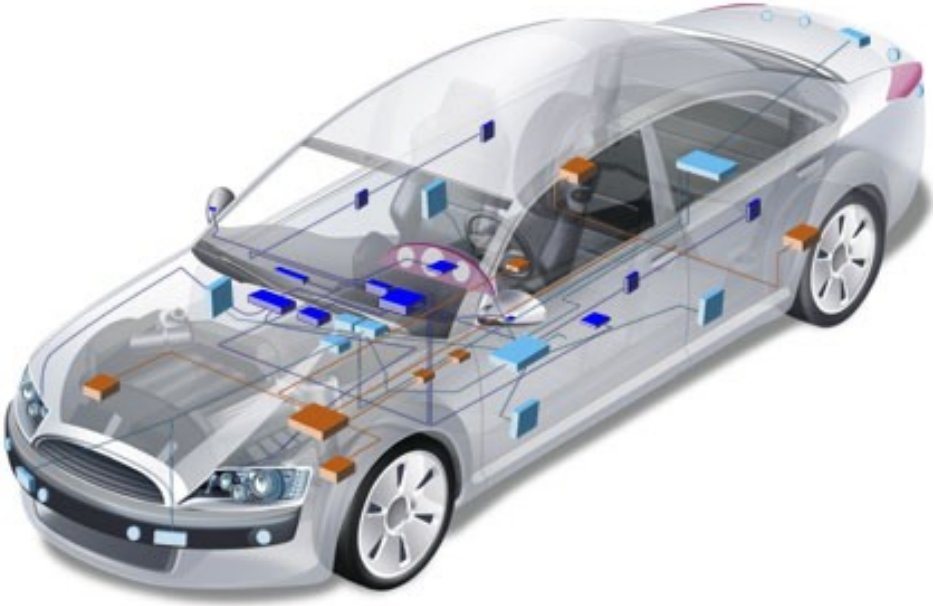
Configuration in Software

- Systems cover all possible functionality
  - e.g., Windows, Open Office, Oracle, SAP myERP, Photoshop
- Specialized software and software for embedded systems increasingly important
  - Mobile divices, sensor networks, automotive systems, consumer electronics, smart cards, ubiquitious computing
  - 98% of all CPUs in embedded devices [2000]
  - Resource constraints, heterogeneous hardware

Database Systems

- Increasing data volumes
- Resource constraints in embedded environments

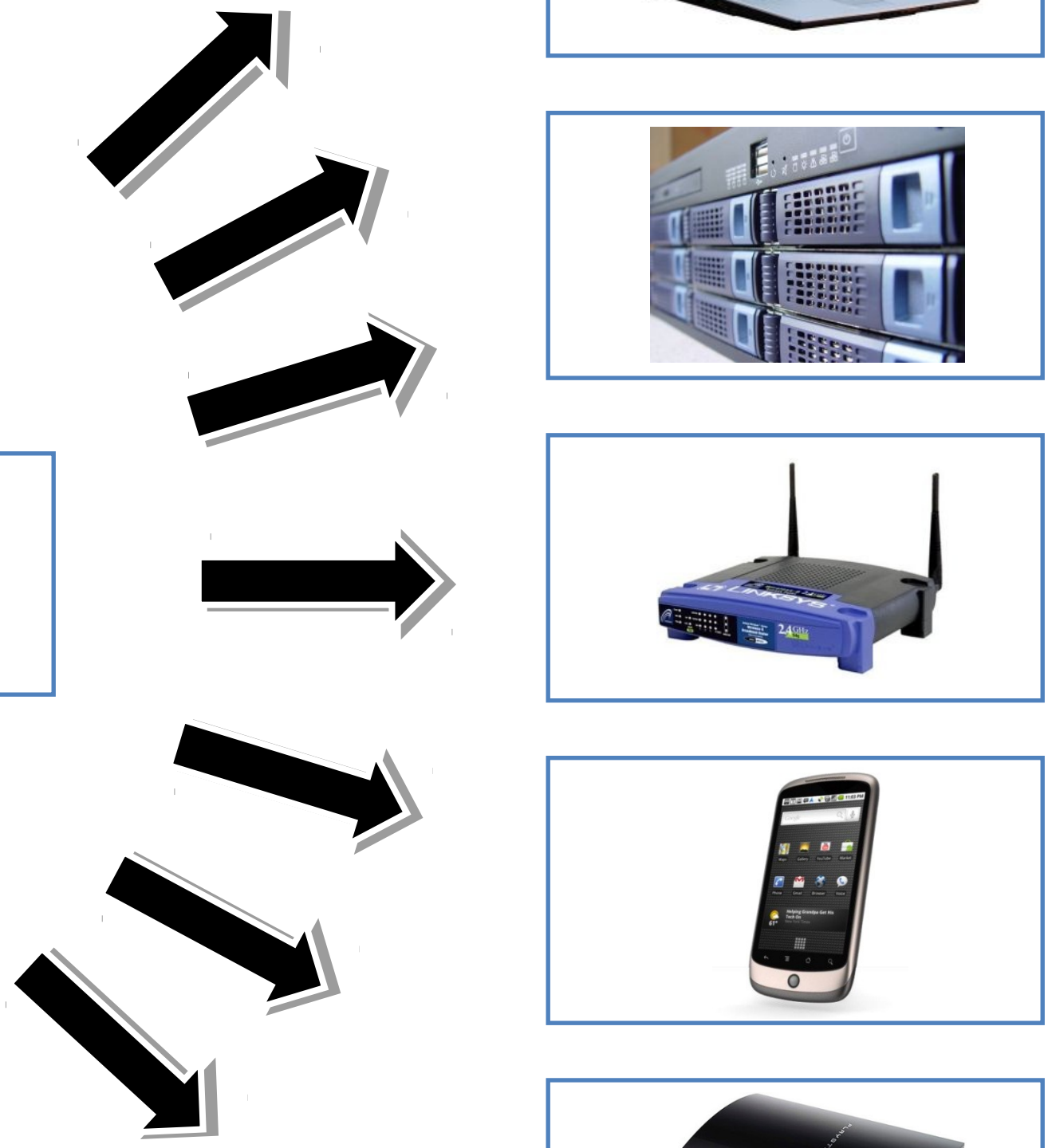




Database Engine
SMART CARD

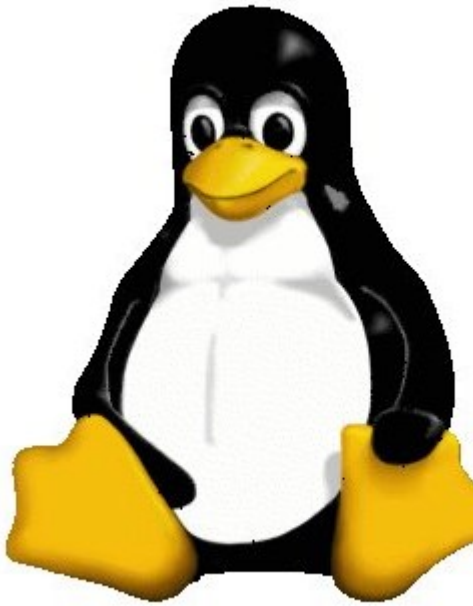
Printer Firmware

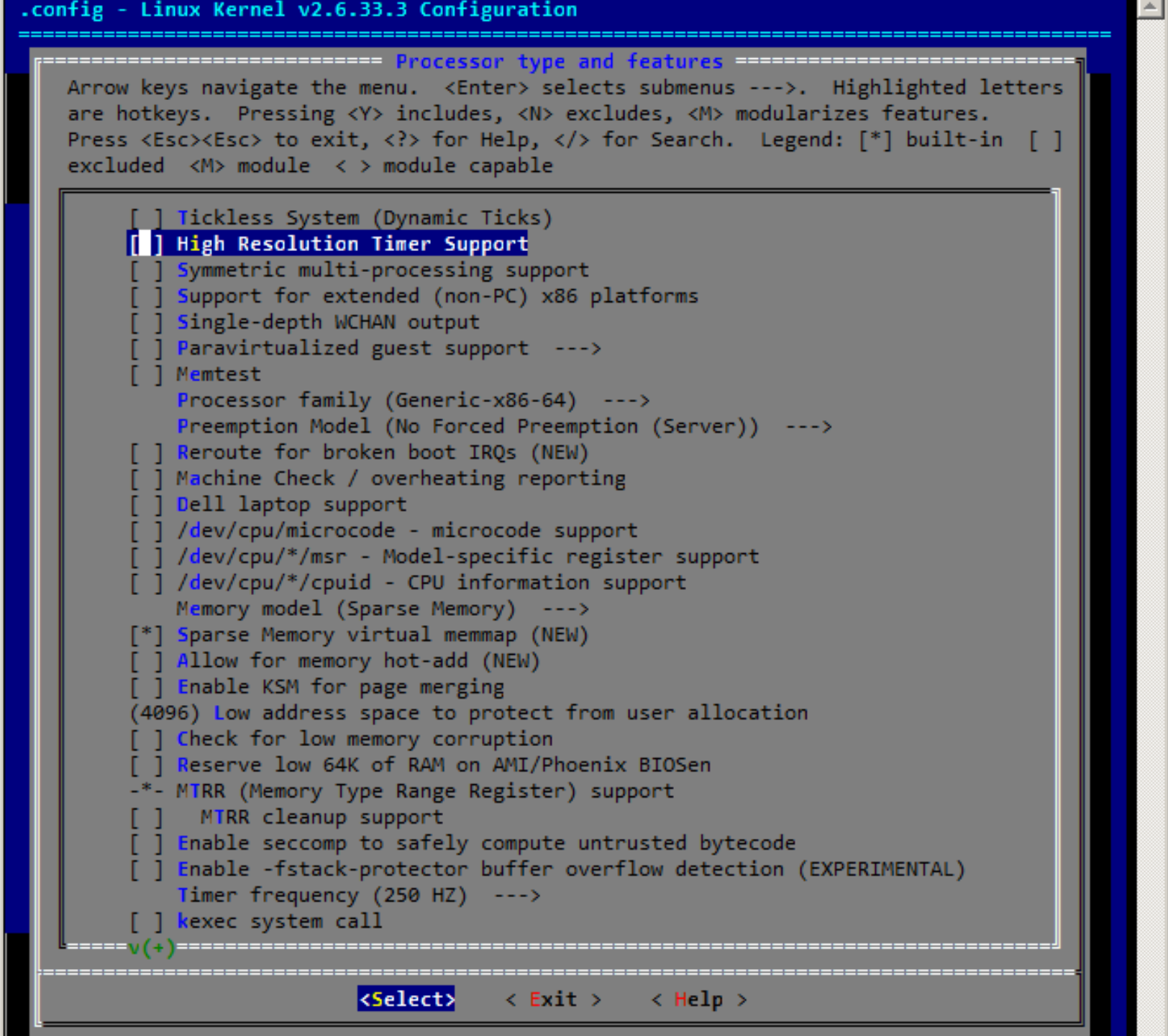
Linux Kernel



Linux kernel

- ~6 000 000 Lines of C code
- Highly configurable
  - > 10.000 configuration options! (x86, 64bit, ⋯)
  - Most source code is “optional”




.config - Linux Kernel v2.6.33.3 Configuration
Processor type and features
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
[ ] Tickless System (Dynamic Ticks)
[ ] High Resolution Timer Support
[ ] Symmetric multi-processing support
[ ] Support for extended (non-PC) x86 platforms
[ ] Single-depth WCHAN output
[ ] Paravirtualized guest support --->
[ ] Memtest
Processor family (Generic-x86-64) --->
Preemption Model (No Forced Preemption (Server)) --->
[ ] Reroute for broken boot IRQs (NEW)
[ ] Machine Check / overheating reporting
[ ] Dell laptop support
[ ] /dev/cpu/microcode - microcode support
[ ] /dev/cpu/*/msr - Model-specific register support
[ ] /dev/cpu/*/cpuid - CPU information support
Memory model (Sparse Memory) --->
[*] Sparse Memory virtual memmap (NEW)
[ ] Allow for memory hot-add (NEW)
[ ] Enable KSM for page merging
(4096) Low address space to protect from user allocation
[ ] Check for low memory corruption
[ ] Reserve low 64K of RAM on AMI/Phoenix BIOSen
-*- MTRR (Memory Type Range Register) support
[ ] MTRR cleanup support
[ ] Enable seccomp to safely compute untrusted bytecode
[ ] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
Timer frequency (250 HZ) --->
[ ] kexec system call
v(+)
<Select> < Exit > < Help >

2013 XTS STANDARD FWD $44,995†

Standard Equipment

PRICE DETAILS +

SELECT DRIVE TYPE

- Front Wheel Drive — MSRP Starting at $44,995**
- All Wheel Drive — MSRP Starting at $51,835**

SELECT TRIM

COMPARE SIDE-BY-SIDE »

Standard	MSRP Starting at $44,995**
Luxury	MSRP Starting at $49,610**
Premium	MSRP Starting at $54,505**
Platinum	MSRP Starting at $59,080**

« Select a Different Vehicle

NEXT STEP »

EXTERIOR: Front Back Side INTERIOR: Driver Side

Some selected options and accessories may not be shown on photos.

VEGETARIAN

WHICH WICH WOULD YOU LIKE?

- ○ TRIPLE CHEESE MELT
- ○ ELVIS WICH (PB, Honey & Banana)
- ○ TOMATO & AVOCADO
- ○ BLACK BEAN PATTY
- ● HUMMUS & BELL PEPPERS

CHOOSE YOUR BREAD

○ WHITE ● WHEAT

CHOOSE YOUR CHEESE (Optional)

○ AMERICAN ○ SWISS ○ PROVOLONE
○ CHEDDAR ● PEPPER JACK ○ MOZZARELLA

How Would You Like Your WICH Worked?

MUSTARDS
○ Yellow ○ Dijon ○ Honey ● Deli

MAYOS
○ Regular ○ Lite ○ Horseradish ● Spicy

SPREADS & SAUCES
○ BBQ ○ Buffalo ○ Marinara
○ 1000 Island ○ Ranch

ONIONS
● Red ○ Grilled ○ Crispy Strings

VEGGIES
● Lettuce ● Tomato ○ Pickles ● Jalapenos
● Olive Salad ○ Mushrooms ○ Sauerkraut
○ Coleslaw ○ Bell Peppers

OILS & SPICES
○ Oil ○ Vinegar
● Salt ● Pepper ○ Oregano ○ Parmesan

EXTRAS (.75¢ Each)
○ Bacon ○ Avocado ○ Pickle (Whole)
○ More Meat ○ More Cheese

Software Product Lines in Industry

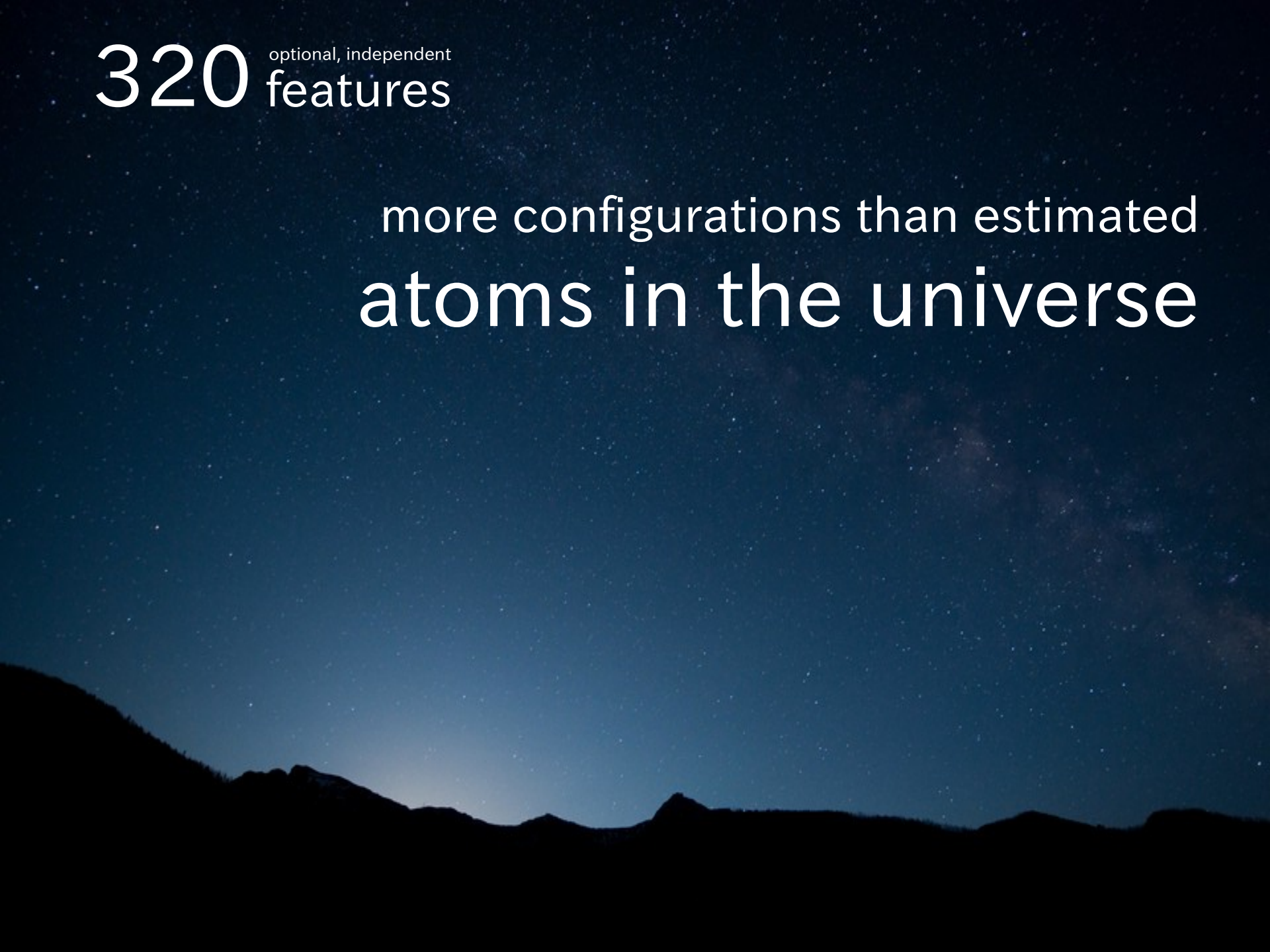
Boeing
Bosch Group
Cummins, Inc.
Ericsson
General Dynamics
General Motors
Hewlett Packard
Lockheed Martin
Lucent
NASA
Nokia
Philips
Siemens
...

CHALLENGES

33 optional, independent features

a unique configuration for every

person on this planet


320 optional, independent features
more configurations than estimated
atoms in the universe

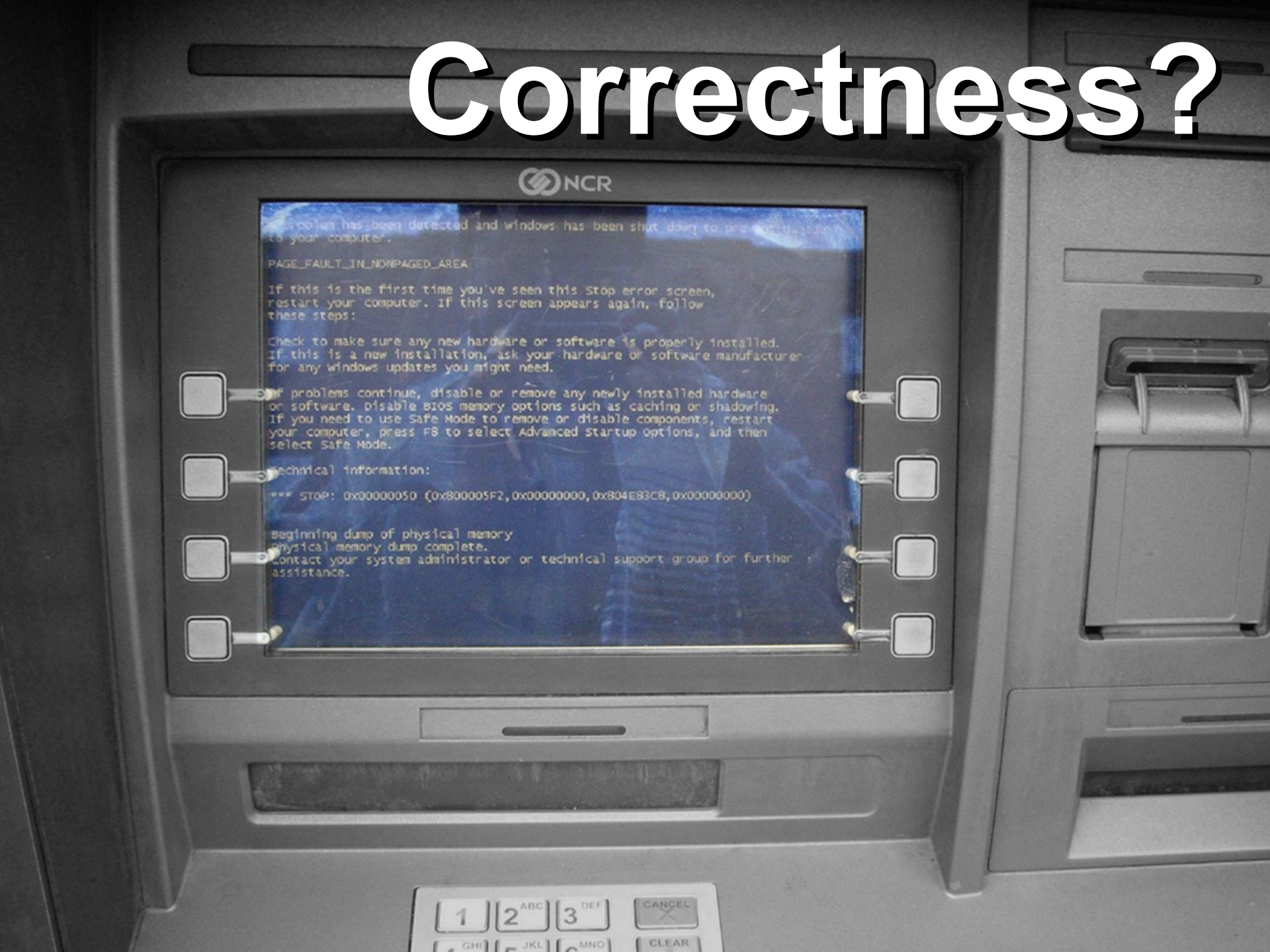
2000 Features

10000 Features

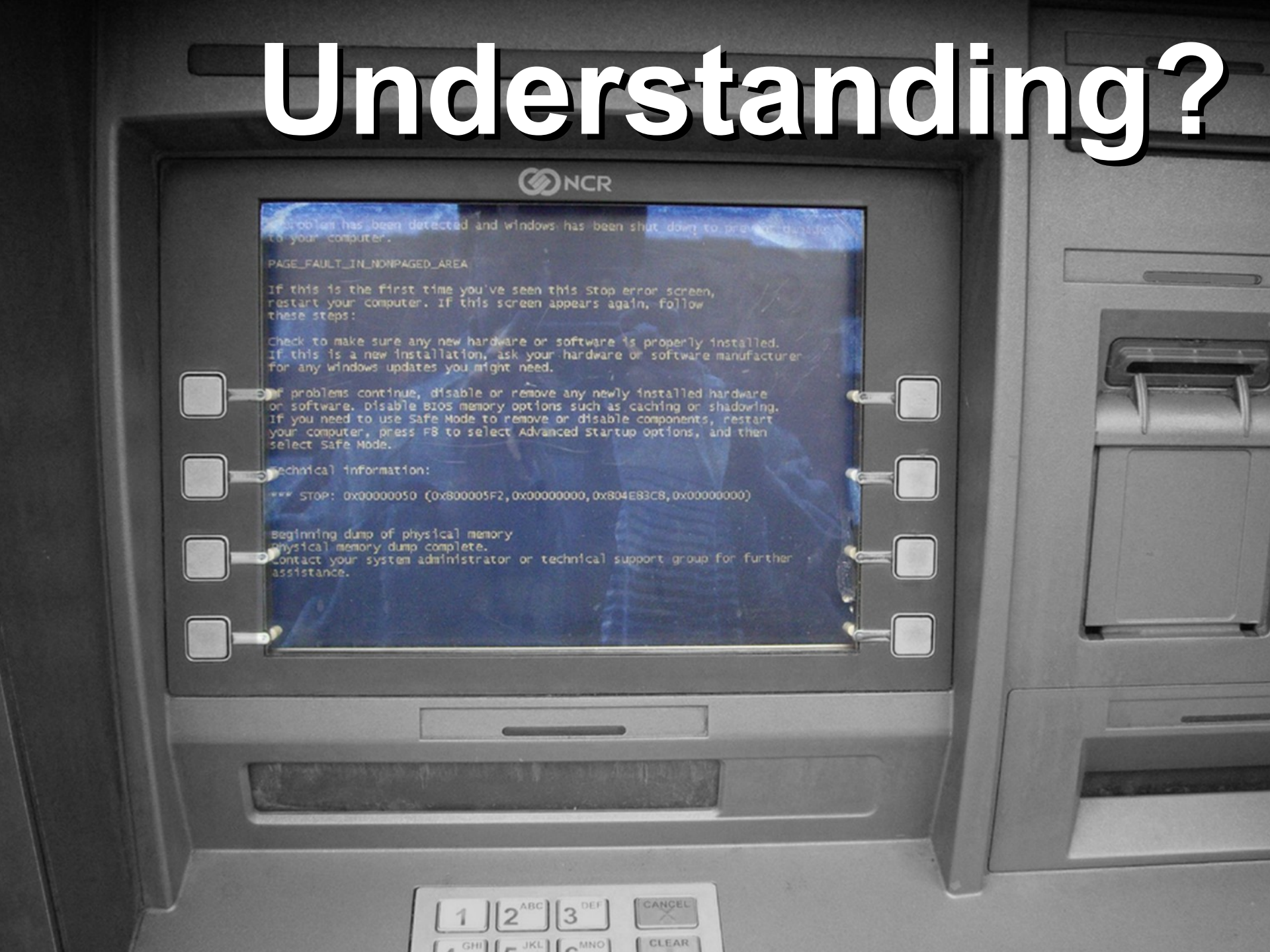
199 Configuration parameters for the optimizer

(1600 total)

Correctness?



Understanding?

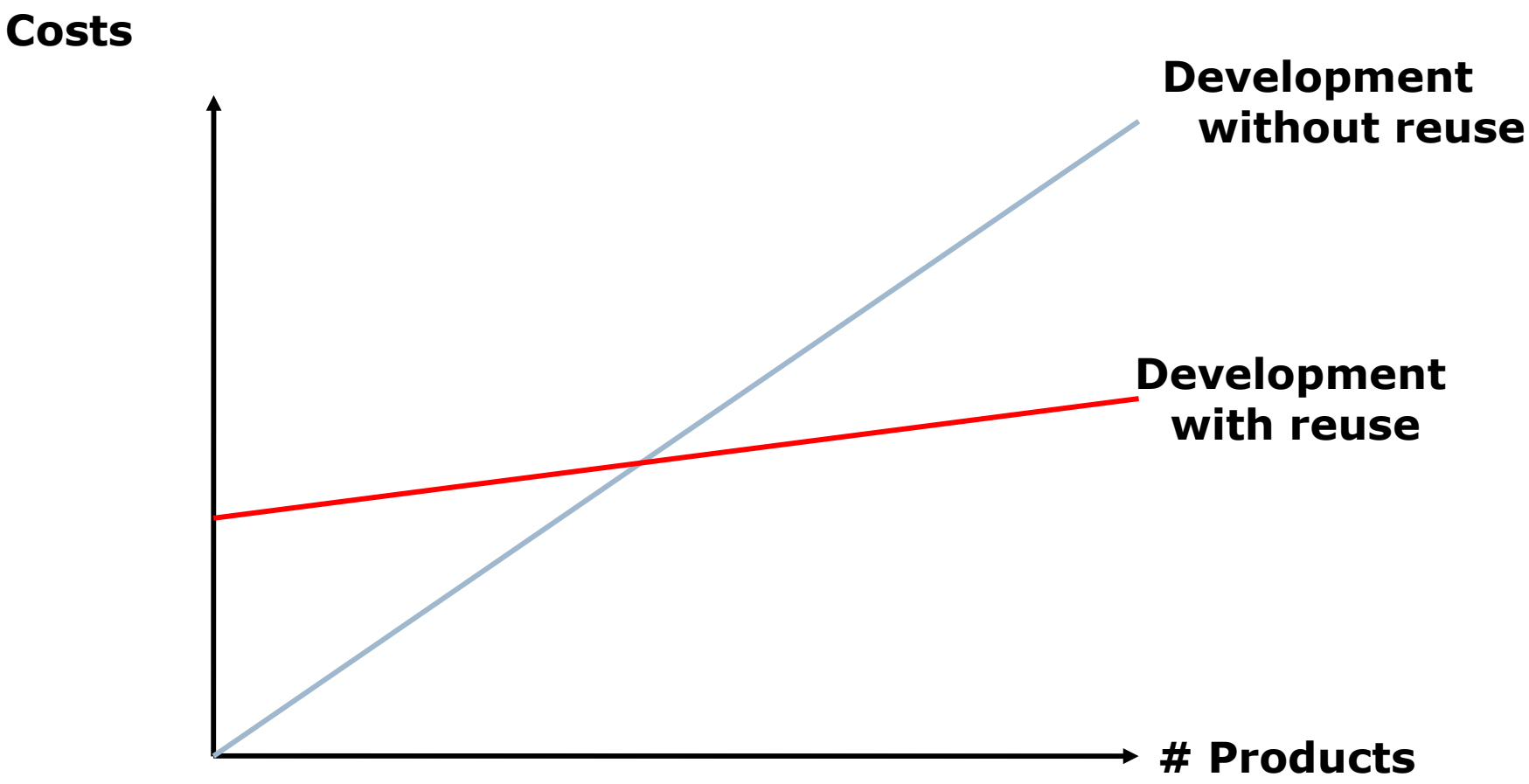


Software Product Lines

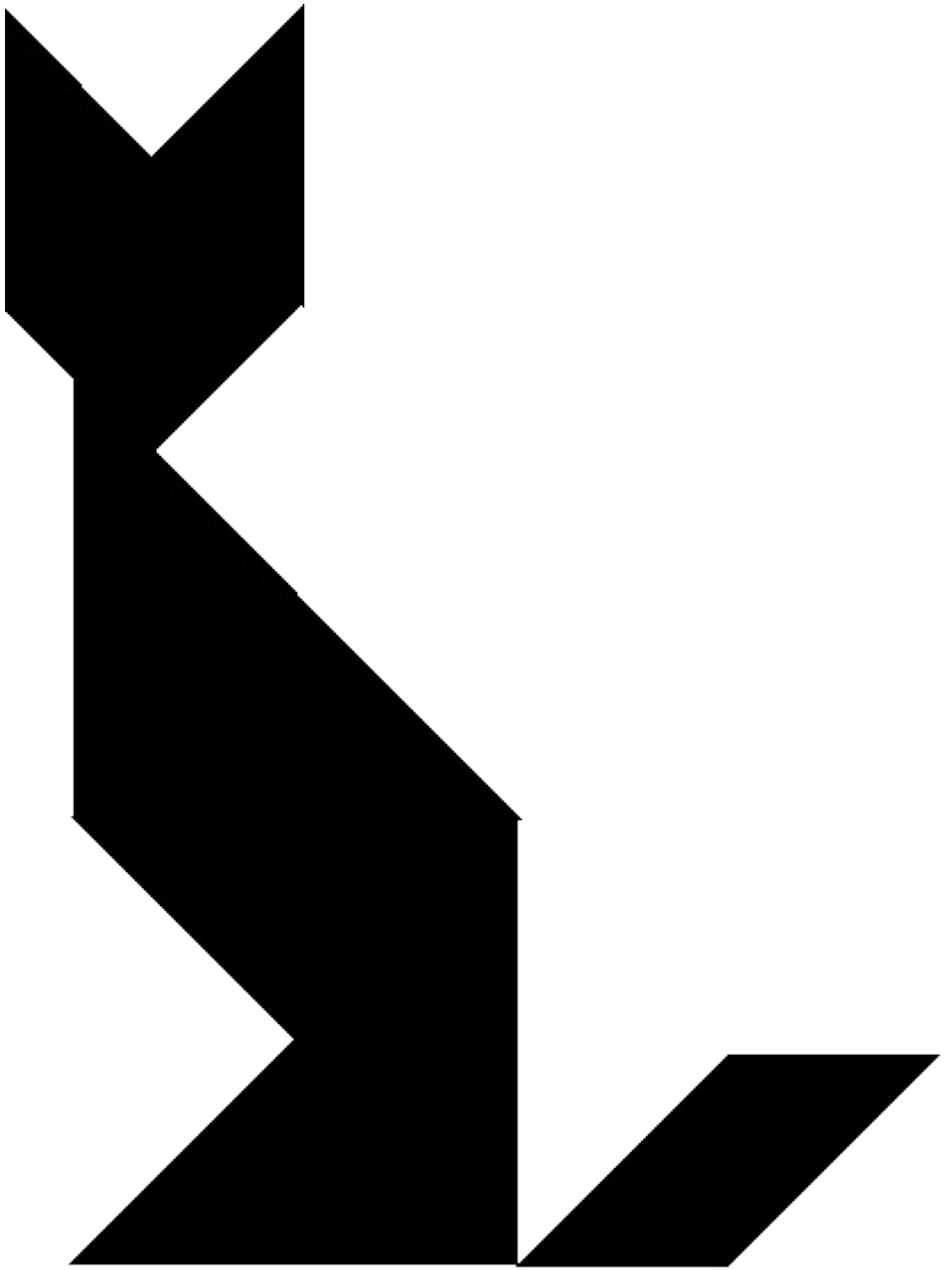
"A software product line is a **set of software-intensive systems** that share a **common, managed set of features** satisfying the specific needs of a **particular market segment** or mission and that are developed from a **common set of core assets** in a prescribed way."

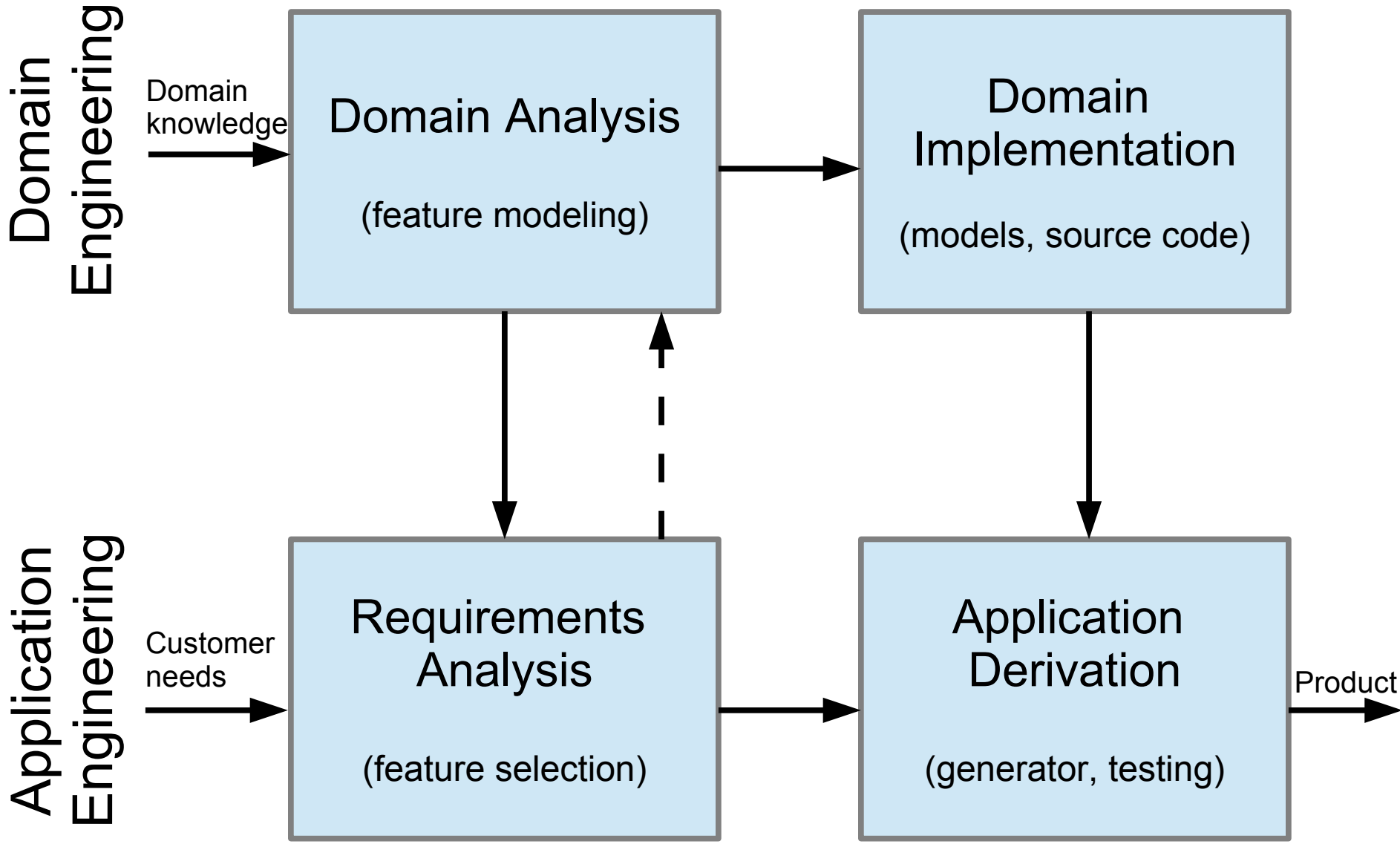
SEI

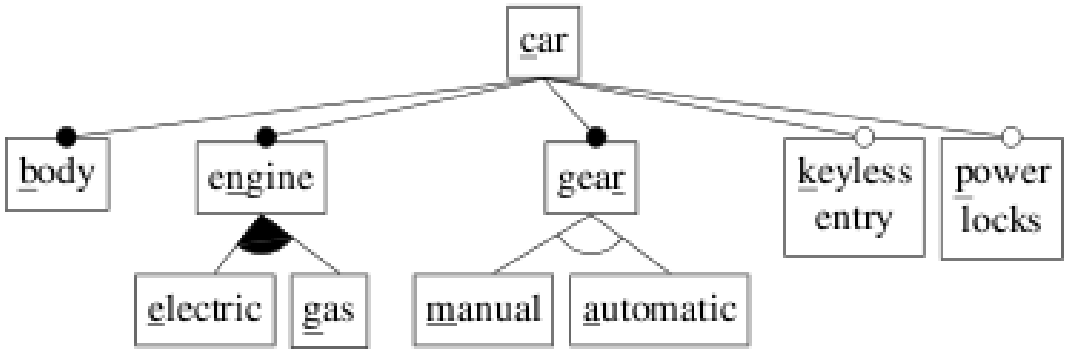
The Promise


Costs
Development without reuse
Development with reuse
Products

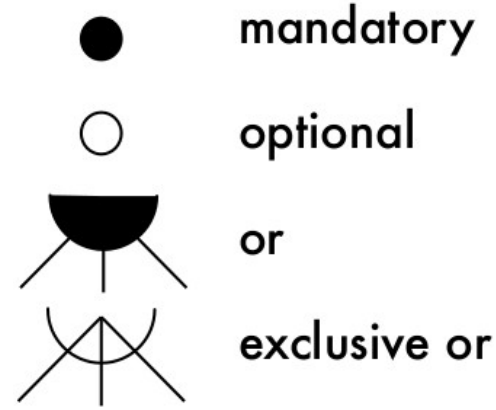
CONQUERING VARIABILITY

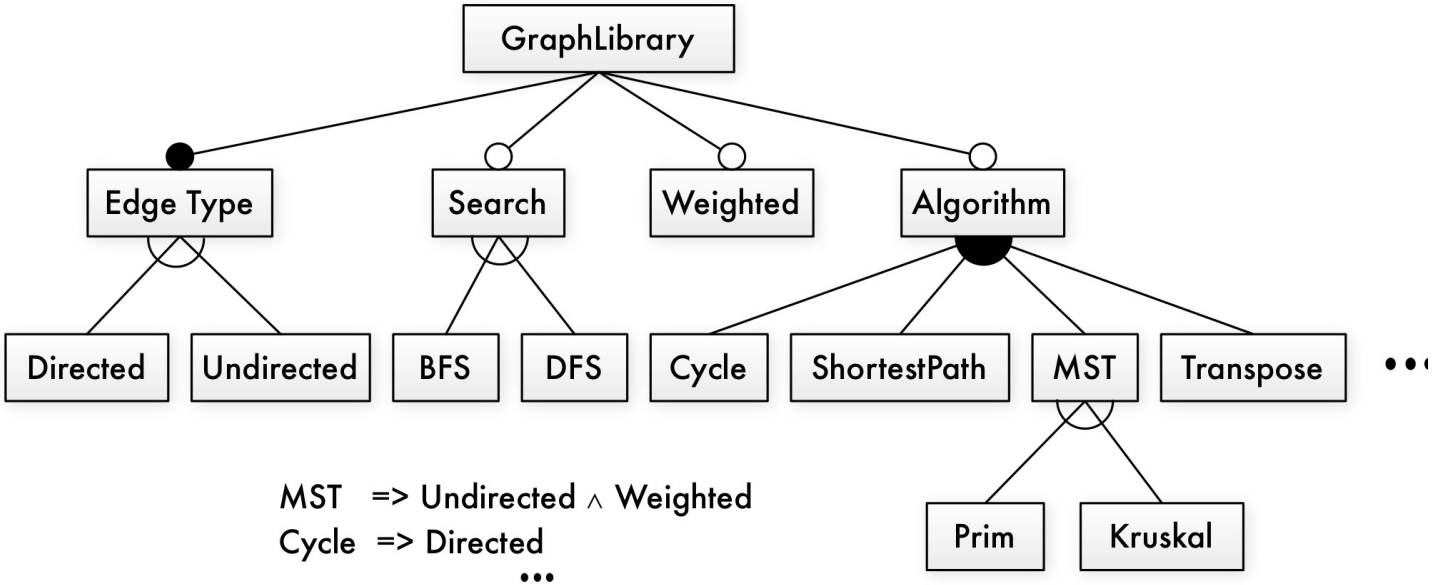



Domain Engineering
Domain knowledge
Domain Analysis
(feature modeling)
Domain Implementation
(models, source code)
Application Engineering
Customer needs
Requirements Analysis
(feature selection)
Application Derivation
(generator, testing)
Product

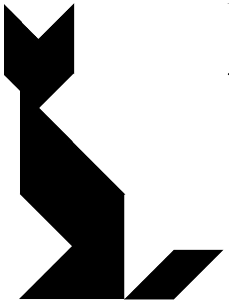


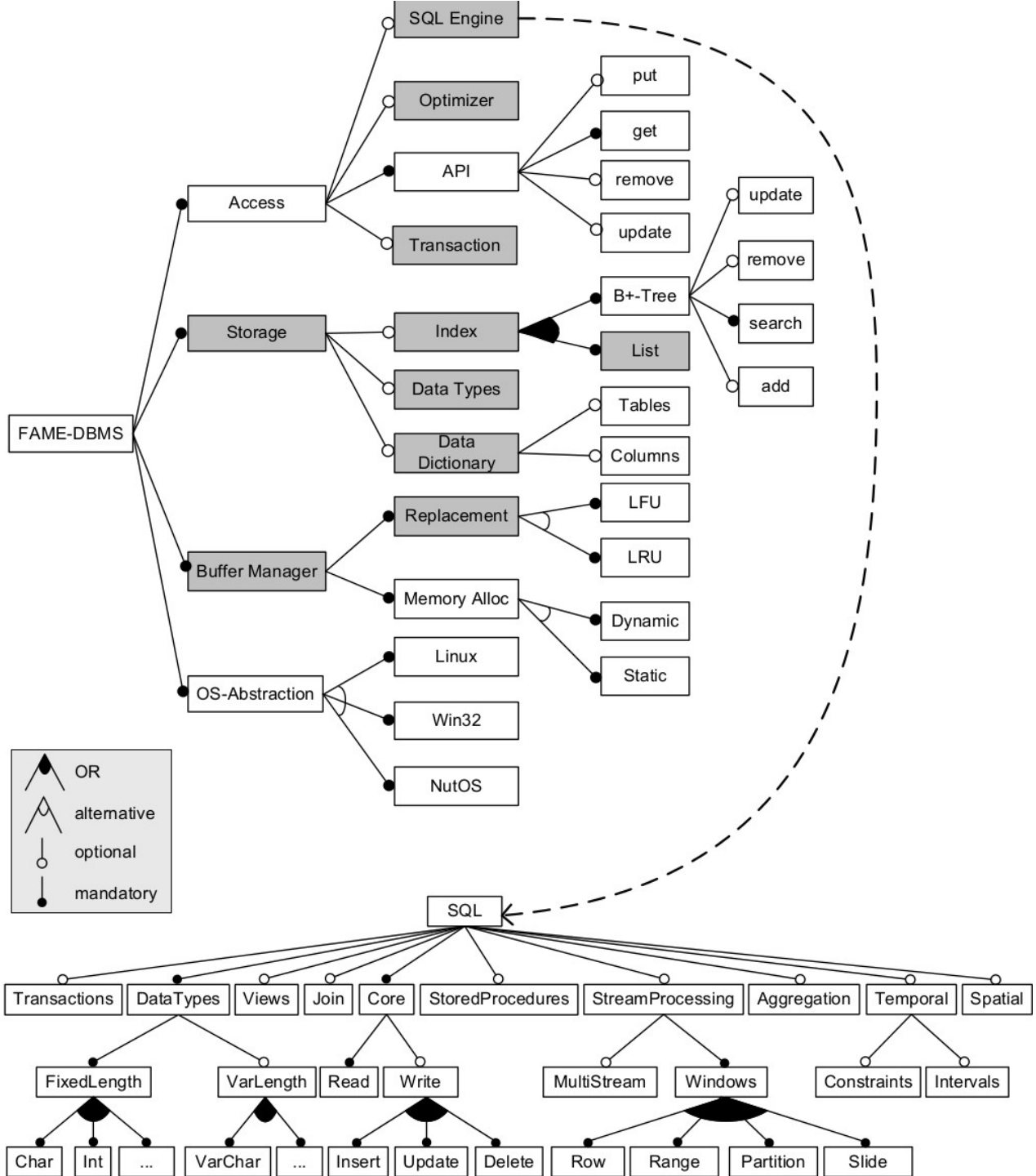
$keyless\_entry \rightarrow power\_locks$





MST => Undirected ∧ Weighted
Cycle => Directed
...




FAME-DBMS
Access
Storage
Buffer Manager
OS-Abstraction
SQL Engine
Optimizer
API
Transaction
put
get
remove
update
Index
B+-Tree
List
update
remove
search
add
Data Types
Data Dictionary
Tables
Columns
Replacement
LFU
LRU
Memory Alloc
Dynamic
Static
Linux
Win32
NutOS
OR
alternative
optional
mandatory
SQL
Transactions
DataTypes
Views
Join
Core
StoredProcedures
StreamProcessing
Aggregation
Temporal
Spatial
FixedLength
VarLength
Read
Write
MultiStream
Windows
Constraints
Intervals
Char
Int
...
VarChar
...
Insert
Update
Delete
Row
Range
Partition
Slide

Linux Kernel v2.6.33.3 Configuration

File Edit Option Help

Option

- TestA
- General setup
  - RCU Subsystem
  - [x] Control Group support
  - [ ] Configure standard kernel features (for small systems)
  - Kernel Performance Events And Counters
  - GCOV-based kernel profiling
- [x] Enable loadable module support
- Enable the block layer (NEW)
  - IO Schedulers
- Processor type and features
  - [ ] Paravirtualized guest support
- Power management and ACPI options
  - [x] ACPI (Advanced Configuration and Power Interface) Support
  - [ ] SFI (Simple Firmware Interface) Support
  - CPU Frequency scaling
  - Memory power savings
- Bus options (PCI etc.)
  - [x] PCCard (PCMCIA/CardBus) support
  - [x] Support for PCI Hotplug
- Executable file formats / Emulations
- [x] Networking support
  - Networking options
    - [x] Network packet filtering framework (Netfilter)
      - Core Netfilter Configuration
      - [ ] IP virtual server support
      - IP: Netfilter Configuration
      - IPv6: Netfilter Configuration
    - [ ] The DCCP Protocol (EXPERIMENTAL)
    - [ ] The SCTP Protocol (EXPERIMENTAL)
    - [ ] The TIPC Protocol (EXPERIMENTAL)
    - [ ] Distributed Switch Architecture support
    - [x] QoS and/or fair queueing
    - Network testing
  - [x] Amateur Radio support
  - [ ] CAN bus subsystem support

Option

- ..
- [x] CPU Frequency scaling
  - [x] Enable CPUfreq debugging
  - [ ] CPU frequency translation statistics
  - Default CPUFreq governor
    - ○ performance
    - ◉ userspace
    - ○ ondemand
    - ○ conservative
  - [x] 'performance' governor
  - 'powersave' governor
  - 'userspace' governor for userspace frequency scaling
  - [x] 'ondemand' cpufreq policy governor
  - [ ] 'conservative' cpufreq governor
  - CPUFreq processor drivers
  - [x] ACPI Processor P-States driver
  - [ ] AMD Opteron/Athlon64 PowerNow!
  - [ ] Intel Enhanced SpeedStep (deprecated)
  - [ ] Intel Pentium 4 clock modulation
  - shared options

'powersave' governor (CPU_FREQ_GOV_POWERSAVE)

CONFIG_CPU_FREQ_GOV_POWERSAVE:

This cpufreq governor sets the frequency statically to the lowest available CPU frequency.

To compile this driver as a module, choose M here: the module will be called cpufreq_powersave.

If in doubt, say Y.

Help

Reset | Open Cfg | Save Cfg | Open Eqn | Save Eqn | DB Table | Form

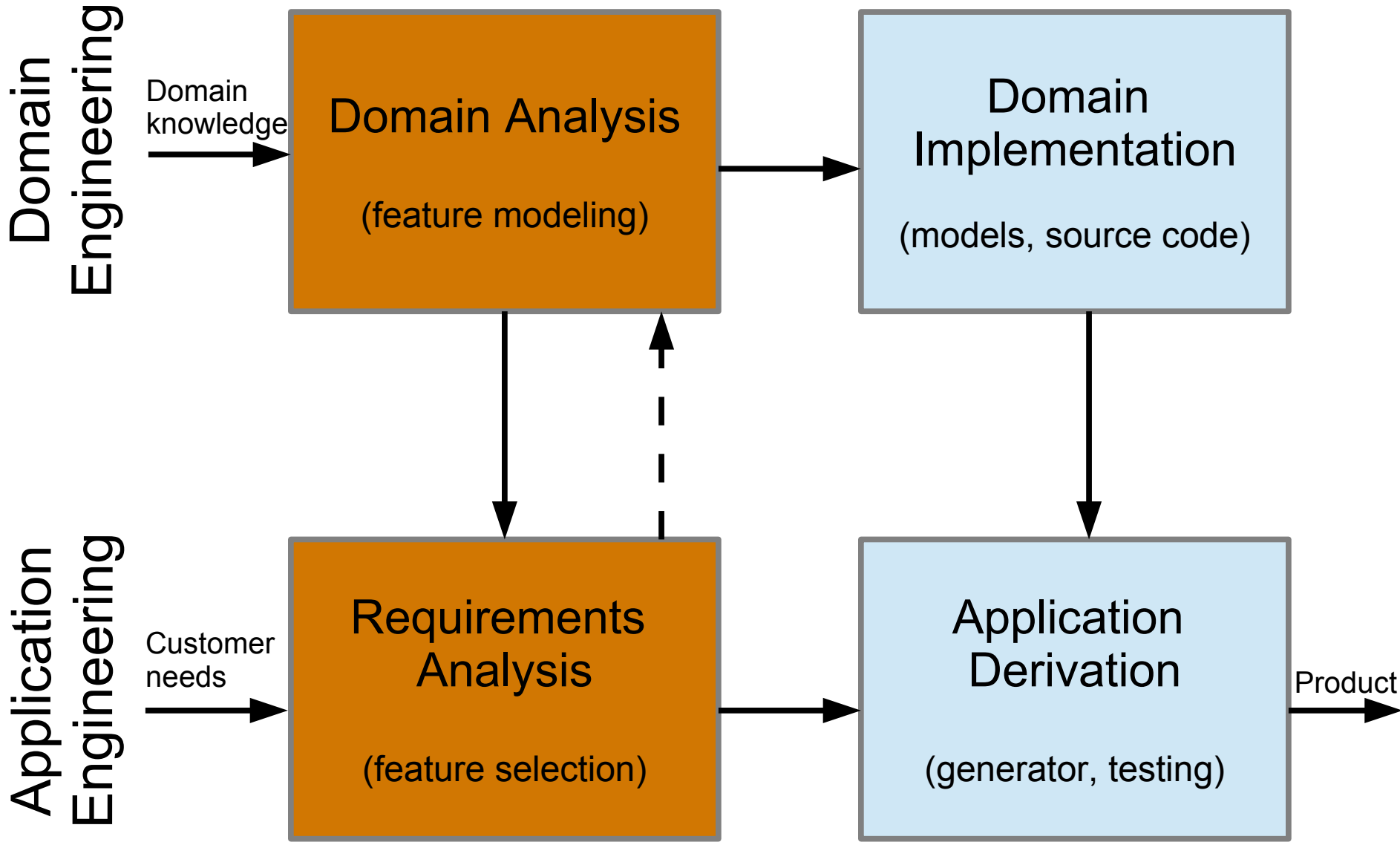
SPL

OS
- ◉ Win
- ○ Unix

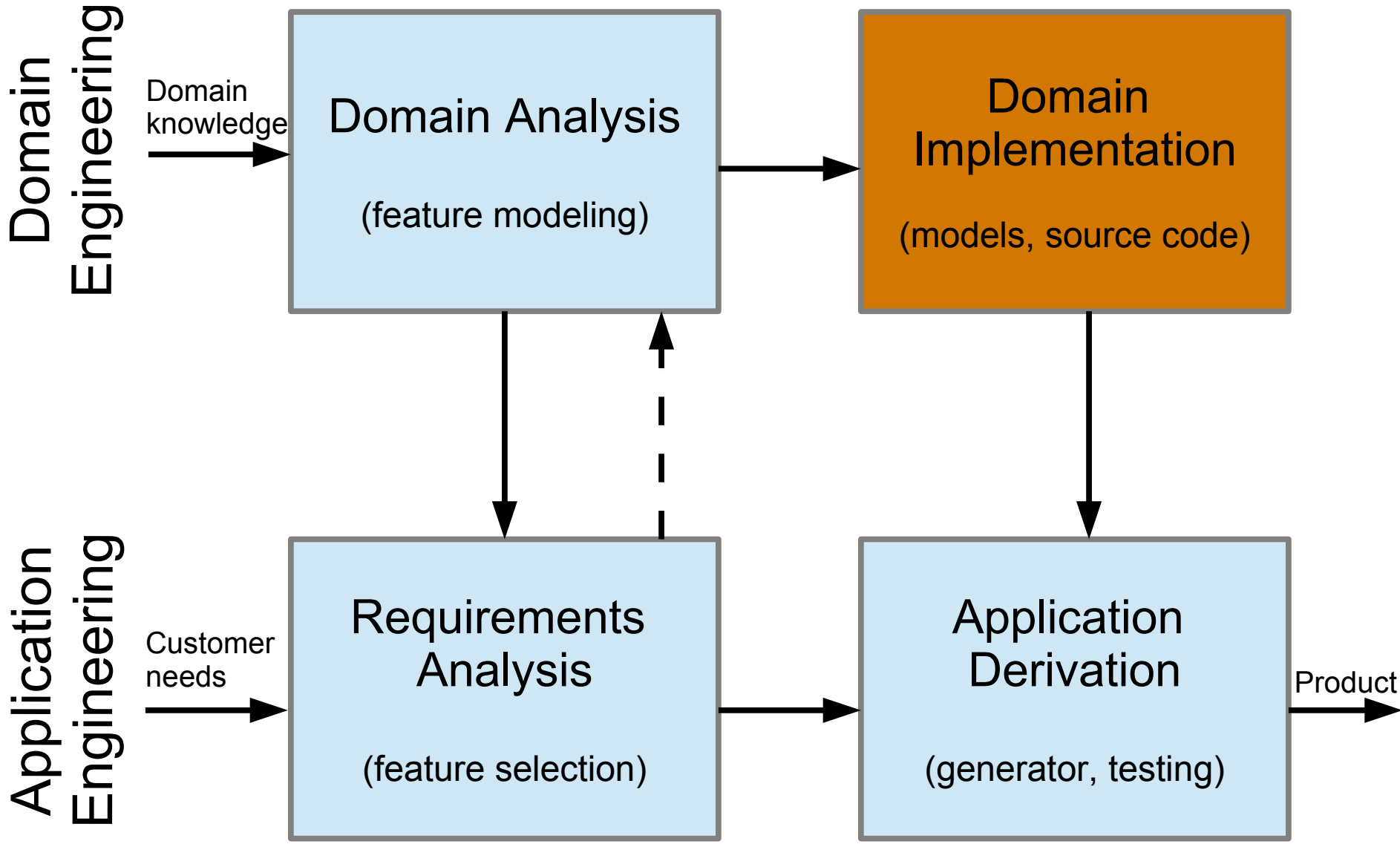
[x] DirectX [ ] Transactions

API
- [x] Get
- [ ] Put
- [ ] Delete

DirectX because set by user
Win because (DirectX) implies (Win)
not Unix because ((OS iff (Unix or Win)) and atmostone(Unix,Win))


Domain Engineering
Domain knowledge
Domain Analysis
(feature modeling)
Domain Implementation
(models, source code)
Application Engineering
Customer needs
Requirements Analysis
(feature selection)
Application Derivation
(generator, testing)
Product

IMPLEMENTATION


Domain Engineering
Domain knowledge
Domain Analysis
(feature modeling)
Domain Implementation
(models, source code)
Application Engineering
Customer needs
Requirements Analysis
(feature selection)
Application Derivation
(generator, testing)
Product

Runtime Parameters

Parameters, variables, constants

Parameter

Administrator: C:\Windows\system32\cmd.exe

```
C:\Users\kaestner.INFORMATIK.000>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
  [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

  [drive:][path][filename]
              Specifies drive, directory, and/or files to list.

  /A          Displays files with specified attributes.
  attributes   D  Directories                R  Read-only files
               H  Hidden files               A  Files ready for archiving
               S  System files               I  Not content indexed files
               L  Reparse Points             -  Prefix meaning not
  /B          Uses bare format (no heading information or summary).
  /C          Display the thousand separator in file sizes.  This is the
              default.  Use /-C to disable display of separator.
  /D          Same as wide but files are list sorted by column.
  /L          Uses lowercase.
  /N          New long list format where filenames are on the far right.
  /O          List by files in sorted order.
  sortorder    N  By name (alphabetic)       S  By size (smallest first)
               E  By extension (alphabetic)  D  By date/time (oldest first)
               G  Group directories first    -  Prefix to reverse order
  /P          Pauses after each screenful of information.
```

Parameter –i in grep

```
1   int match_icase;
2
3   int main (int argc, char **argv)
4   {
5     [...]
6     while ((opt = get_nondigit_option (argc, argv, &default_c
7       switch (opt)
8         {
9         [...]
10        case 'i':
11          match_icase = 1;
12          break;
13        }
14    }
15
16
17  static const char *
18  print_line_middle (const char *beg, const char *lim,
19                     const char *line_color, const char *match_color)
20  {
21    [...]
22    if (match_icase)
23      {
24        ibeg = buf = (char *) xmalloc(i);
25        while (--i >= 0)
26          buf[i] = tolower(beg[i]);
```

Global configuration options

```
class Config {
  public static boolean isLogging = false;
  public static boolean isWindows = false;
  public static boolean isLinux = true;
}
class Main {
  public void foo() {
    if (isLogging)
      log(“running foo()“);
    if (isWindows)
      callWindowsMethod();
    else if (isLinux)
      callLinuxMethod();
    else
      throw RuntimeException();
}
```

Propagating Parameters

Database.java (W:\work\db\BerkeleyDb\src\com\sleepycat\je) - GVIM

File Edit Tools Syntax Buffers Window Help

```
 * the doc templates in the doc_src directory.
 */
public Sequence openSequence(Transaction txn,
                             DatabaseEntry key,
                             SequenceConfig config)
    throws DatabaseException {

    checkEnv();
    DatabaseUtil.checkForNullDbt(key, "key", true);
    checkRequiredDbState(OPEN, "Can't call Database.openSequence:");
    checkWritable("openSequence");
    trace(Level.FINEST, "Database.openSequence", txn, key, null, null);

    return new Sequence(this, txn, key, config);
}

/**
 * Javadoc for this public method is generated via
 * the doc templates in the doc_src directory.
 */
public void removeSequence(Transaction txn, DatabaseEntry key)
    throws DatabaseException {

    delete(txn, key);
}

public synchronized Cursor openCur

    throws DatabaseException {
```

Database.java (W:\work\db\BerkeleyDb\src\com\sleepycat\je) - GVIM

File Edit Tools Syntax Buffers Window Help

```
 * the doc templates in the doc_src directory.
 */
public Sequence openSequence(Transaction txn,
                             DatabaseEntry key,
                             SequenceConfig config)
    throws DatabaseException {
```

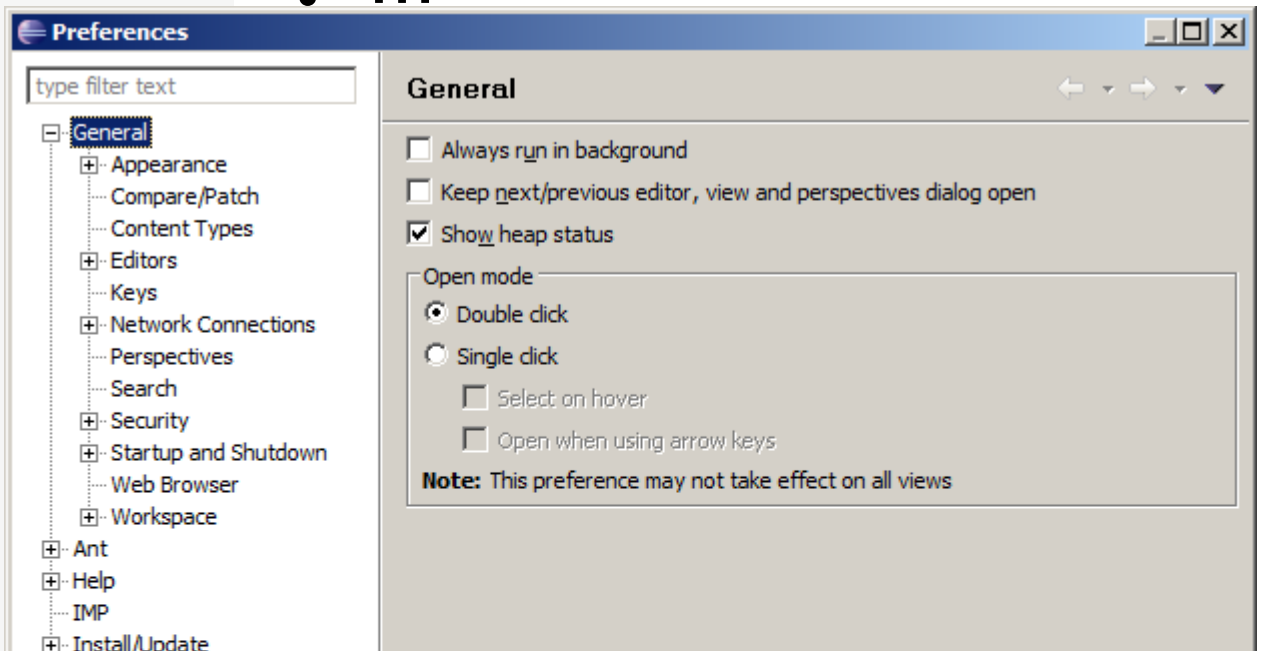
Selecting configurations

httpd.conf -- win32 Apache

Building a Web Server, for Windows

```
Listen 80
ServerRoot "/www/Apache2"
DocumentRoot "/www/webroot"

ServerName localhost:80
ServerAdmin admin@localhost

ServerSignature On
ServerTokens Full

DefaultType text/plain
AddDefaultCharset ISO-8859-1

UseCanonicalName Off

HostnameLookups Off

ErrorLog logs/error.log
LogLevel error

PidFile logs/httpd.pid

Timeout 300

KeepAlive On
MaxKeepAliveRequests 100
```

- Command line parameters
- Config file
- User dialog
- Source code
- …



Graph Library

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

```
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();
    }
  }
}
```

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

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

199 Configuration parameters for the optimizer

(1600 total)

Limitations

- Variable code scattered in entire program
  - Hard to test in isolation
- Global variables or long parameter lists

- All code always included
  - Binary size, memory consumption
  - Performance
  - Unused functionality as attack vector
- Changes at runtime or load-time?

Limitations

- Variable code scattered in entire program
  - Hard to test in isolation
- Global variables or long parameter lists

design pattern

- All code always included
  - Binary size, memory consumption
  - Performance
  - Unused functionality as attack vector
- Changes at runtime or load-time?

compile-time if

Preprocessors

conditional compilation

#ifdef

A Performance Analysis

What's the performance problem?

```
public foo() {
  …
  if (logger.inDebug()) {
    logger.debug("We have " + conn + "connections.");
  }
}
```

Seems minor…
but if this performance gain on 1000 servers means we need 1 less machine, we could be saving a a lot of money

Berkeley DB

```
#include "db_int.h"

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
```

Preprocessor in Java?

- No native preprocessor
- Some compilers support conditional compilation at statement level

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

        void main() {
                System.out.println("immer");
                if (DEBUG) {
                        System.out.println("debug info");
                        printDetails();
                }
        }
        void printDetails() { ... }
}
```

- External tools

Munge

- Simple external preprocessor for Java
- Originally developed for Swing 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

Configurable Graph Library

```
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]*/
```

2000 Features

10000 Features

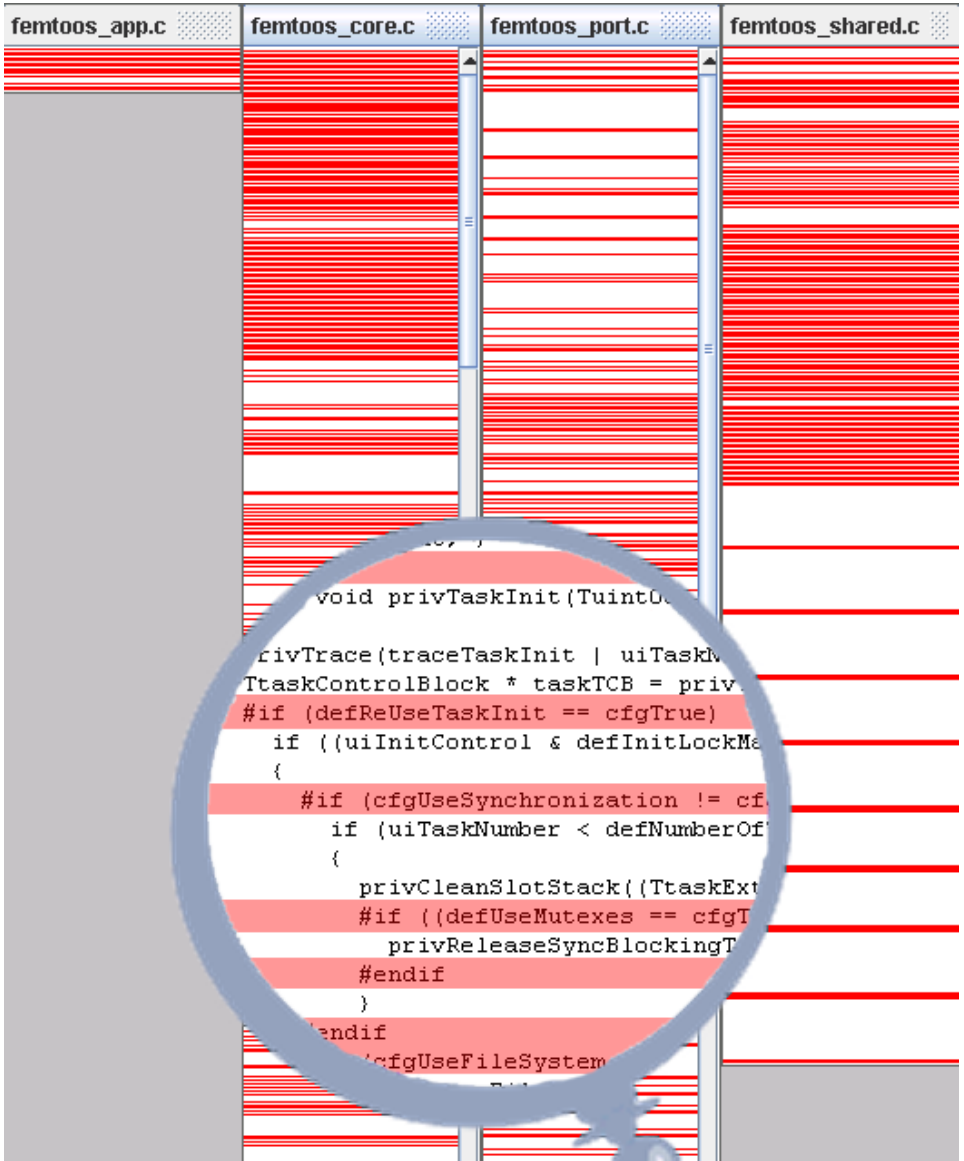
Discussion

- Compile-time configuration
- Can remove arbitrary code before compilation
- Simple programming model

- Bad reputation
  - Scattered code
  - Error prone
  - Hard to understand
  - Invites neglecting design
  - Hinder tool support

```
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();
    }
}
```

Femto OS


femtoos_app.c
femtoos_core.c
femtoos_port.c
femtoos_shared.c
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

```
  int n = NUM2INT(num);
#ifndef FEAT_WINDOWS
  w = curwin;
#else
  for (w = firstwin; w != NULL;
      w = w->w_next, --n)
#endif
    if (n == 0)
      return window_new(w);
```

```
  int n = NUM2INT(num);
#ifndef FEAT_WINDOWS
  w = curwin;
#else
  for (w = firstwin; w != NU
      w = w->w_next, --n)
#endif
    if (n == 0)
      return window_new(w);
```

```
int put_eol(fd)
    FILE *fd;
{
  if (
#ifdef USE_CRNL
    (
#ifdef MKSESSION_NL
      !mksession_nl &&
#endif
      (putc('\r', fc) < 0)) ||
#endif
      (putc('\n', fd) < 0))
    return FAIL;
  return OK;
}
```

Error Prone

```
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
}
```

Error Prone

```
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
}
```

A matter of scale

ApplicationSession

StandardSession

ServerSession

- Example: Session expiration in the Apache Tomcat Server

SessionInterceptor

StandardManager

StandardSessionManager

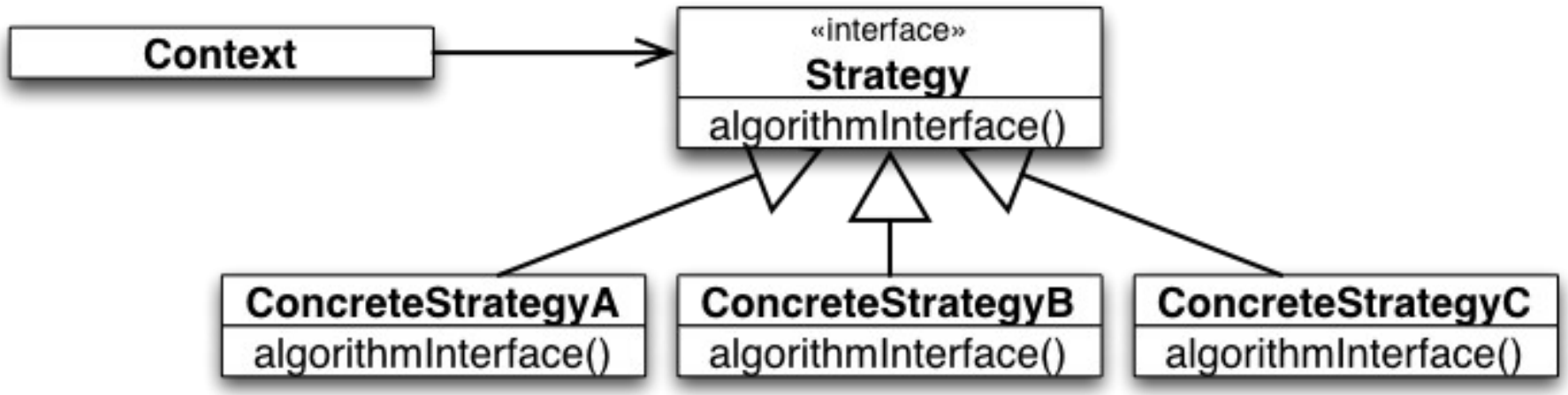
ServerSessionManager

Design Pattern and Frameworks

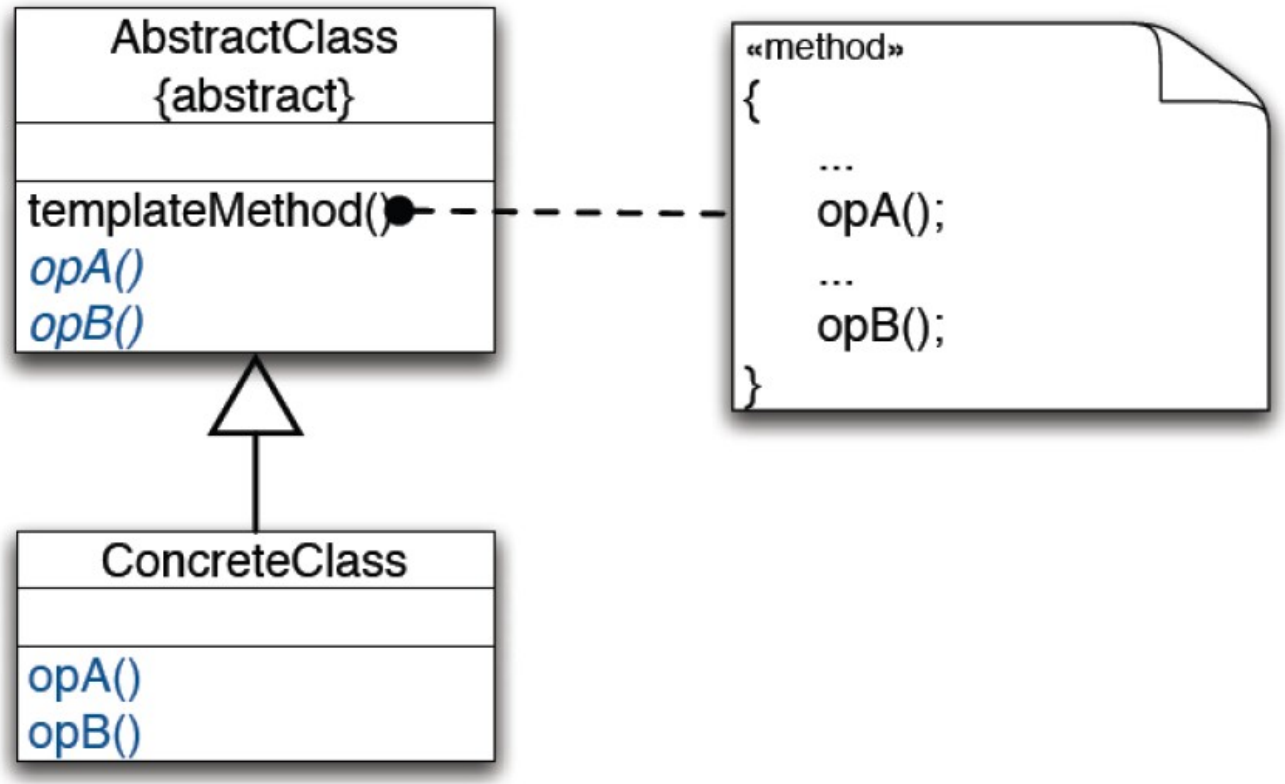
best practice

separation

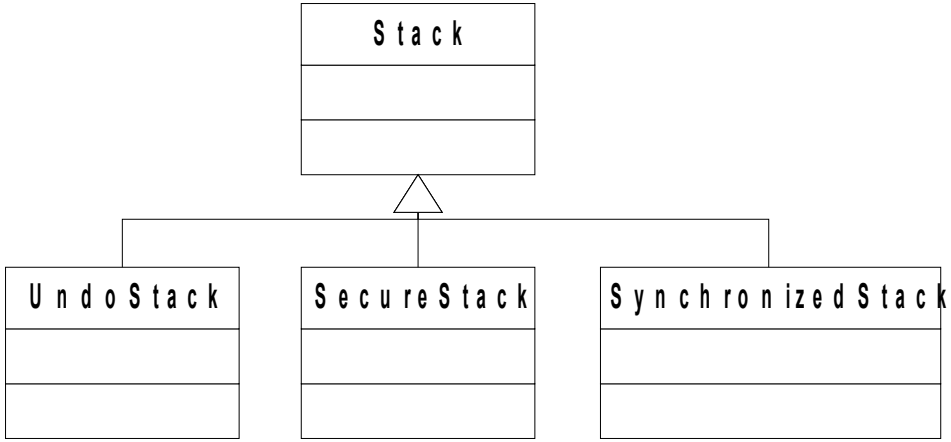
Strategy Pattern



Template Method Pattern

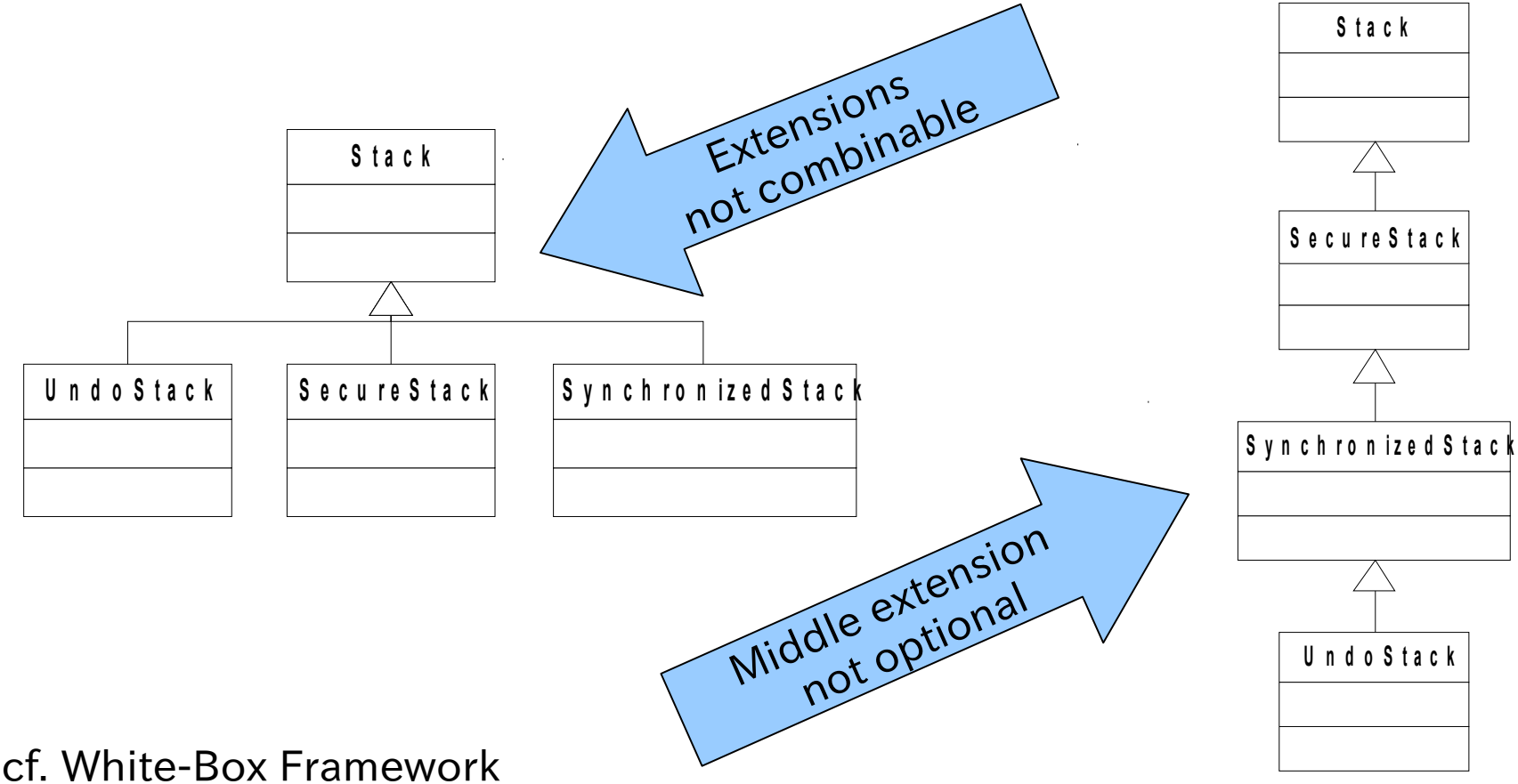


Inheritance for Variability



modular, but inflexible

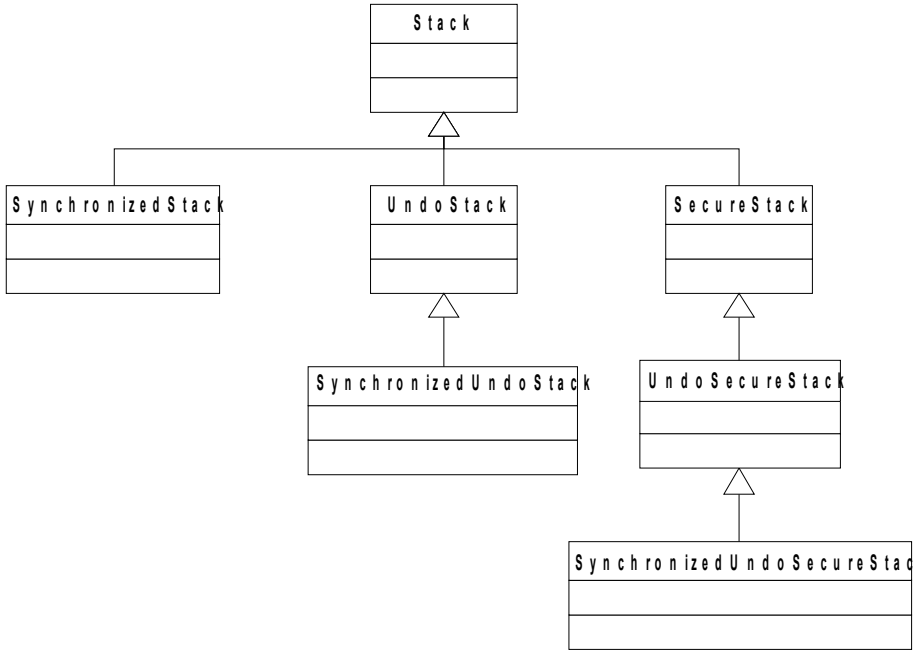
Inflexible Extension Mechanism

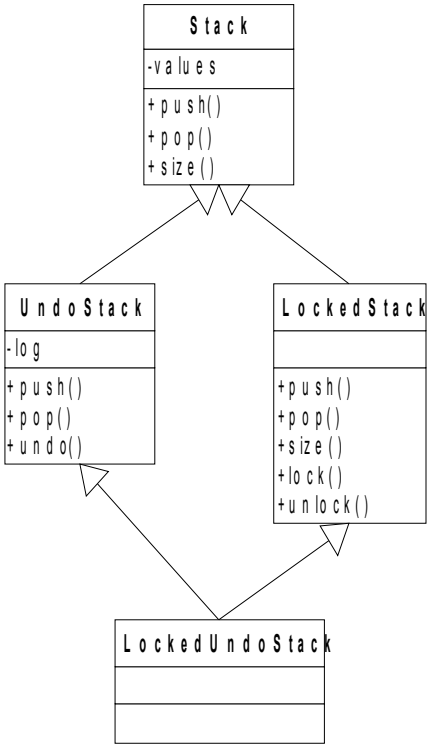


cf. White-Box Framework

Work arounds?

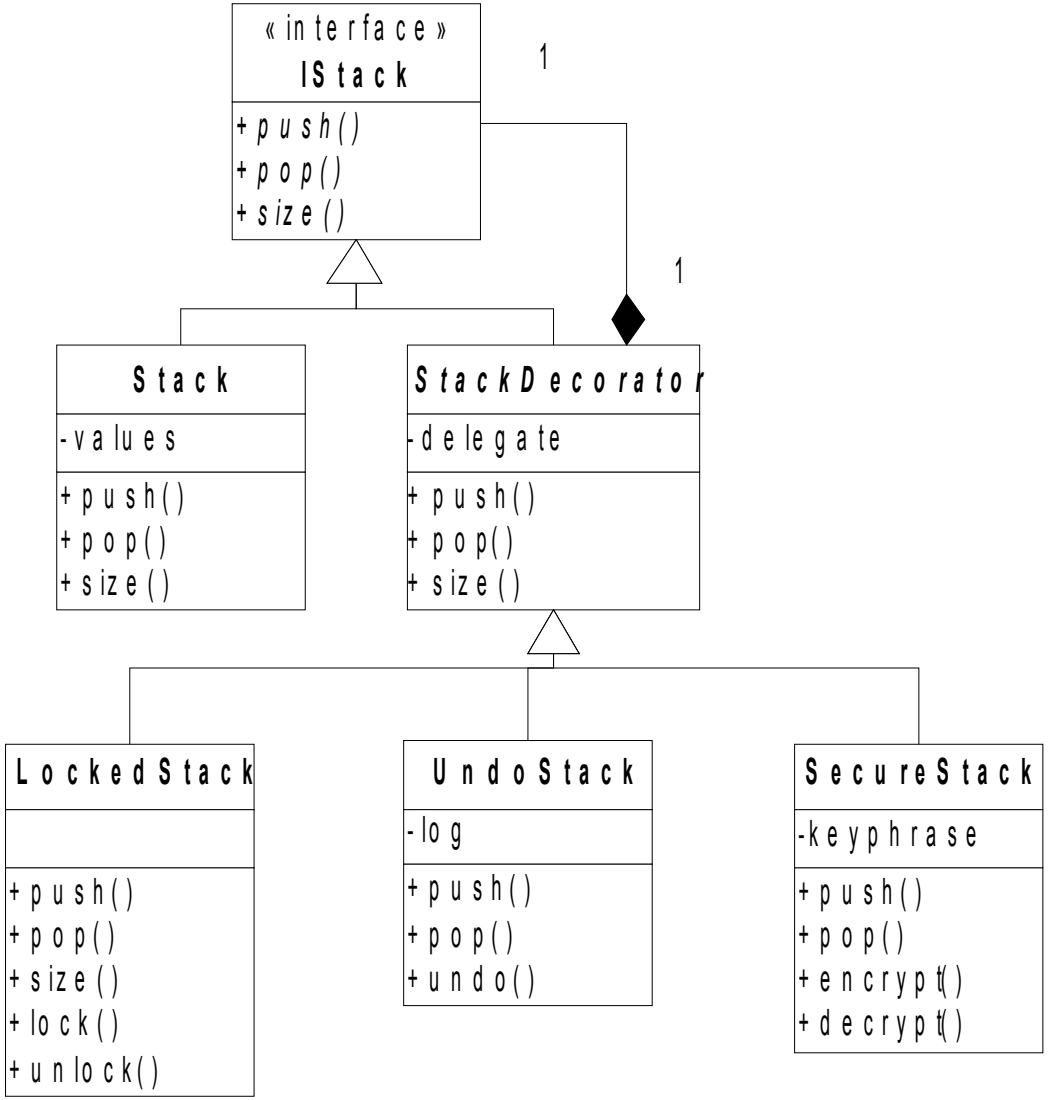
- Combining inheritance hierarchies
  - Combinatorical explosion
  - Massive code replication





- Multiple inheritance
  - Diamond problem

Decorator Pattern



Frameworks

- Set of abstract and concrete classes
- Reusable solutions of a set of problems
- Planned hot spots, extended by plug-ins

- Extension mechanisms often based on design patterns
  - Strategy
  - Template method
  - Observer

Discussion

- Separated, modular extensions
  - Compile and test plug-ins separately
- Compile-time/load-time configuration
- Well understood in practice
- Hard to design, hard to evolve
- Small runtime overhead
- Suited for coarse-grained extensions

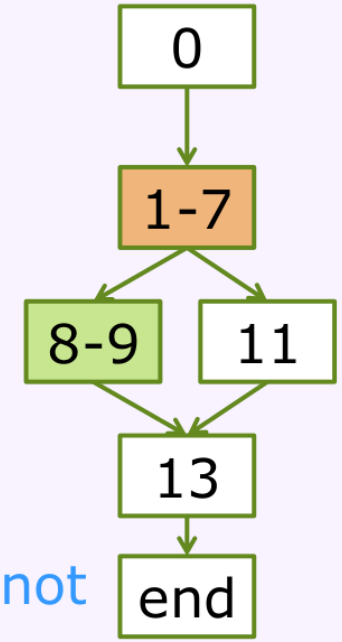
ANALYSIS

Example: Interrupt Problem

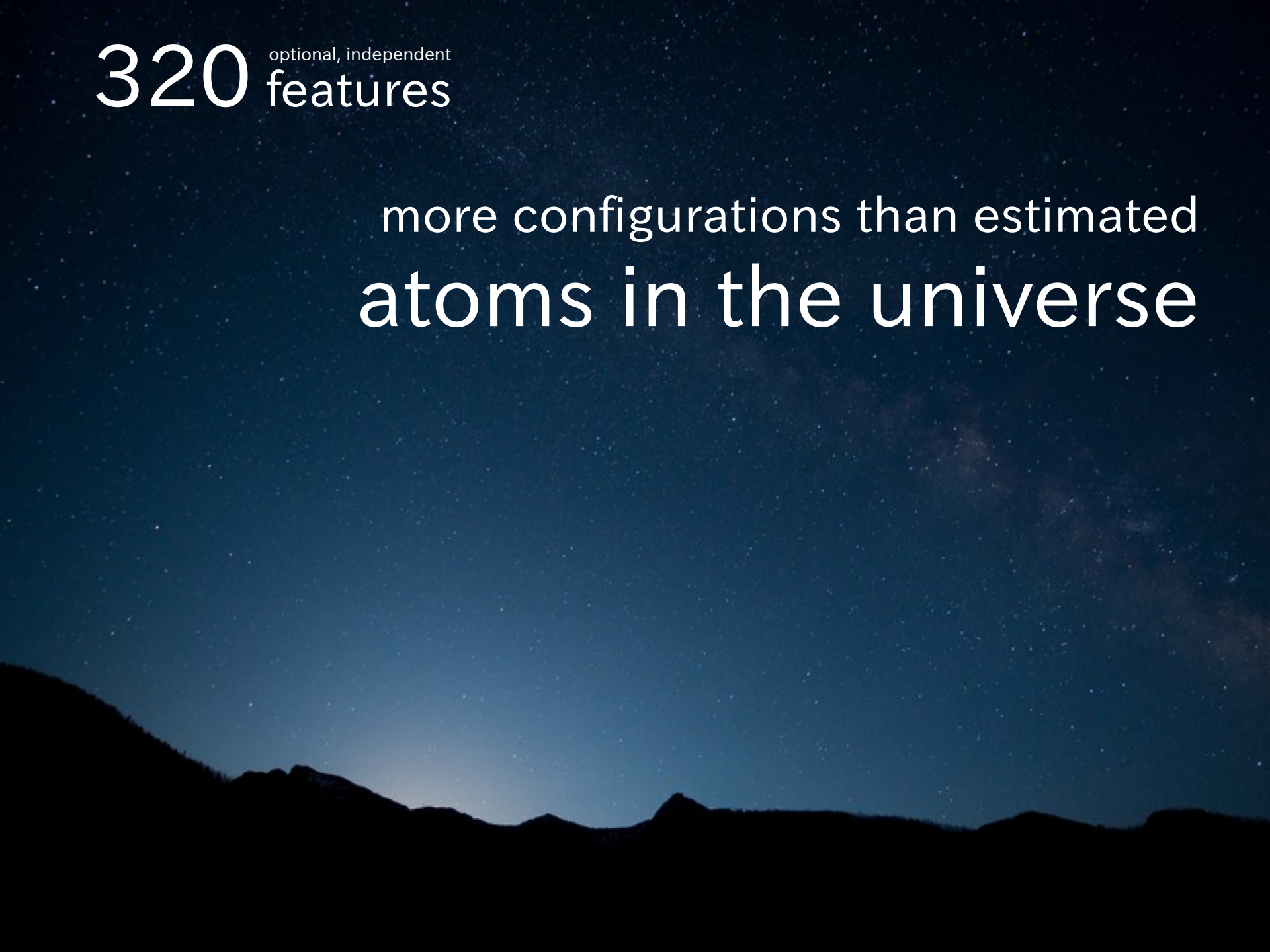
```
1.  int foo() {
2.      unsigned long flags;
3.      int rv;
4.      save_flags(flags);
5.      cli();
6.      rv = dont_interrupt();
7.      if (rv > 0) {
8.          do_stuff();
9.          restore_flags();
10.     } else {
11.         handle_error_case();
12.     }
13.     return rv;
14. }
```

Abstraction (before statement)

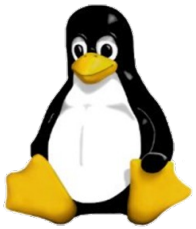
2-4: enabled
5: enabled
6: disabled
7: disabled
8: disabled
9: disabled
11: disabled
13: unknown



Error: did not reenable interrupts on some path


320 optional, independent features
more configurations than estimated
atoms in the universe

From HP to Linux



2000 Features
Inhouse configuration
100 Printers (Product Map)
30 New Printers per Year

10000 Features
End user configures
2^{10000} Configurations

$$\forall\, p \in PL\ .\ \ldots$$

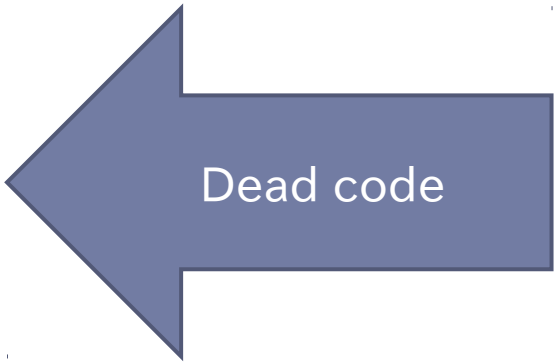
Dead-Code Detection

```
line 1
#ifdef A
line 2
    #ifndef A
    line 3
    #endif
line 4
#elif defined(X)
line 5
#else
line 6
#endif
```

true

A

A ^ not A



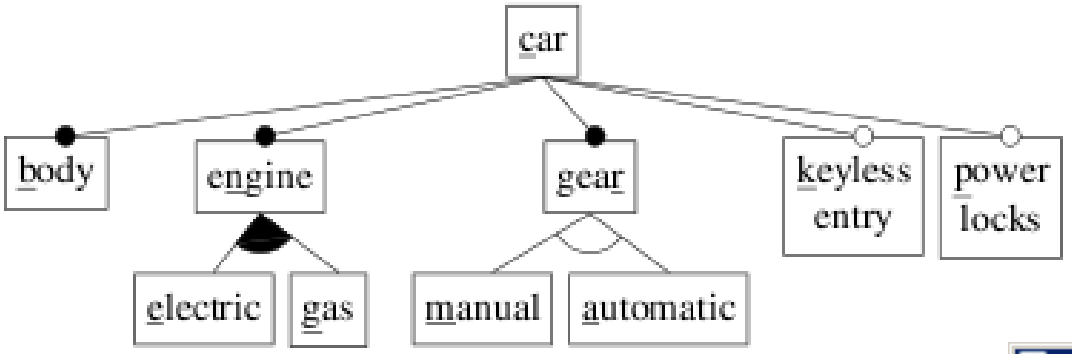
A

not A ^ X

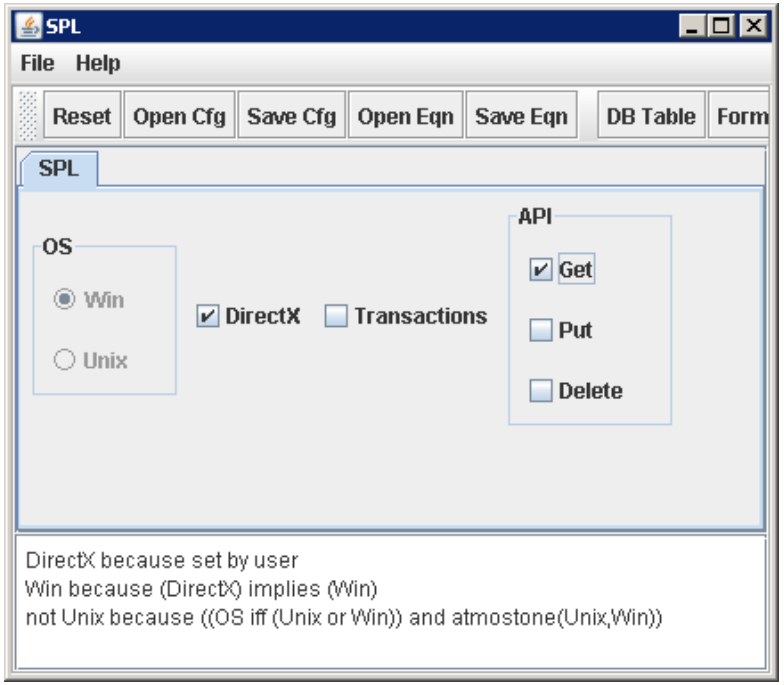
not A ^ not X

Analysis:
SAT(pc(block i))

Variability Models



$keyless\_entry \rightarrow power\_locks$



MainWindow

Show Load

Explorer

Features Filestructure

- xenomai
  - examples
    - common
    - native
    - posix
    - rtdm
  - include
  - ksrc
    - arch
      - arm
      - blackfin
      - generic
        - compat.c
        - hal.c
        - nmi.c
      - nios2
      - powerpc
      - x86

Source Code Viewer

timer.h hal.c

```
46  #include <asm/unistd.h>
47  #include <asm/xenomai/hal.h>
48  #ifdef CONFIG_PROC_FS
49  #include <linux/proc_fs.h>
50  #endif /* CONFIG_PROC_FS */
51  #include <stdarg.h>
52
53  MODULE_LICENSE("GPL");
54
55  unsigned long rthal_cpufreq_arg;
56  module_param_named(cpufreq, rthal_
57
58  unsigned long rthal_timerfreq_arg
59  module_param_named(timerfreq, rth
60
61  unsigned long rthal_clockfreq_arg
62  module_param_named(clockfreq, rth
63
64  #ifdef CONFIG_SMP
65  static unsigned long supported_cp
66  module_param_named(supported_cpus
67
68  cpumask_t rthal_supported_cpus;
69  EXPORT_SYMBOL(rthal_supported_cpu
70  #endif /* CONFIG_SMP */
71
72  static struct {
73
74      void (*handler) (void *cookie
```

Feature Model

- Xenomai
  - Nucleus
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_
    - CONFIG_

Colors

ColorAssignment

Explorer

Source Code Viewer

Features File Structure

- Example
  - src
    - dfa.c
    - dfasearch.c
    - dosbuf.c
    - egrep.c
    - fgrep.c
    - grep.c
    - kwsearch.c
    - kwset.c
    - main.c
    - pcresearch.c
    - searchutils.c

main.c

```c
825 }
826
827 static void
828 prline (char const *beg, char const *lim, int sep)
829 {
830   int matching;
831   const char *line_color;
832   const char *match_color;
833
834   if (!only_matching)
835     print_line_head(beg, lim, sep);
836
837   matching = (sep == SEP_CHAR_SELECTED) ^ !!out_invert;
838
839   if (color_option)
840     {
841       line_color  = (  (sep == SEP_CHAR_SELECTED)
842                        ^ (out_invert && (color_option < 0)))
843                      ? selected_line_color  : context_line_color;
844       match_color = (sep == SEP_CHAR_SELECTED)
845                      ? selected_match_color : context_match_color;
846     }
847   else
848     line_color = match_color = NULL; /* Shouldn't be used.  */
849
850   if (   (only_matching && matching)
851       || (color_option  && (*line_color || *match_color)))
852     {
853       /* We already know that non-matching lines have no match (to colorize).  */
854       if (matching && (only_matching || *match_color))
855           beg = print_line_middle(beg, lim, line_color, match_color);
856
857       /* FIXME: this test may be removable.  */
858       if (!only_matching && *line_color)
859           beg = print_line_tail(beg, lim, line_color);
860     }
861
862   if (!only_matching && lim > beg)
863     fwrite (beg, 1, lim - beg, stdout);
```

Pairwise Testing

F1	F2	F3
on	on	on
on	on	off
on	off	on
on	off	off
off	on	on
off	on	off
off	off	on
off	off	off

Cohen et al. *Interaction testing of highly-configurable systems in the presence of constraints.* ISSTA 2007

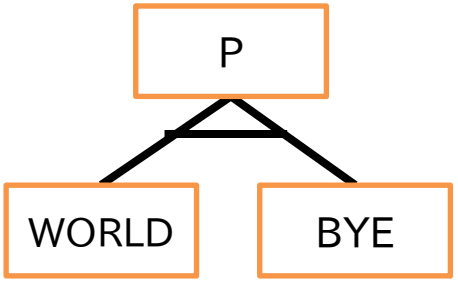
Pairwise Testing

F1	F2	F3
~~on~~	~~on~~	~~on~~
on	on	off
on	off	on
~~on~~	~~off~~	~~off~~
off	on	on
~~off~~	~~on~~	~~off~~
~~off~~	~~off~~	~~on~~
~~off~~	~~off~~	~~off~~

Cohen et al. *Interaction testing of highly-configurable systems in the presence of constraints.* ISSTA 2007

Closed-World Analysis

$\forall$ p $\in$ PL . p is well-typed



(WORLD v BYE) ^
¬ (WORLD ^ BYE)

```
#include <stdio.h>

#ifdef WORLD
char * msg = "Hello␣World\n";
#endif
#ifdef BYE
char * msg = "Bye␣bye!\n";
#endif

main() {
  printf(msg);
}
```

¬ (WORLD ^ BYE)

WORLD v BYE

Specified Variability

Implementation Variability

Other Topics

- Product planning, product-line scoping
- Team organization
- Development and design process (domain engineering)
- More implementation techniques (components, version control systems, feature/aspect-oriented development)
- Product-line analysis

BUY OUR
PRODUCT
(we're on a billboard).

Further Reading

- S. Apel, D. Batory, C. Kästner, and G. Saake. Feature-Oriented Software Product Lines: Concepts and Implementation. Berlin/Heidelberg: Springer-Verlag, 2013.
- K. Czarnecki and U. Eisenecker. Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.
- P. Clements, L. Northrop. *Software Product Lines: Practices and Patterns.* Addison-Wesley, 2002

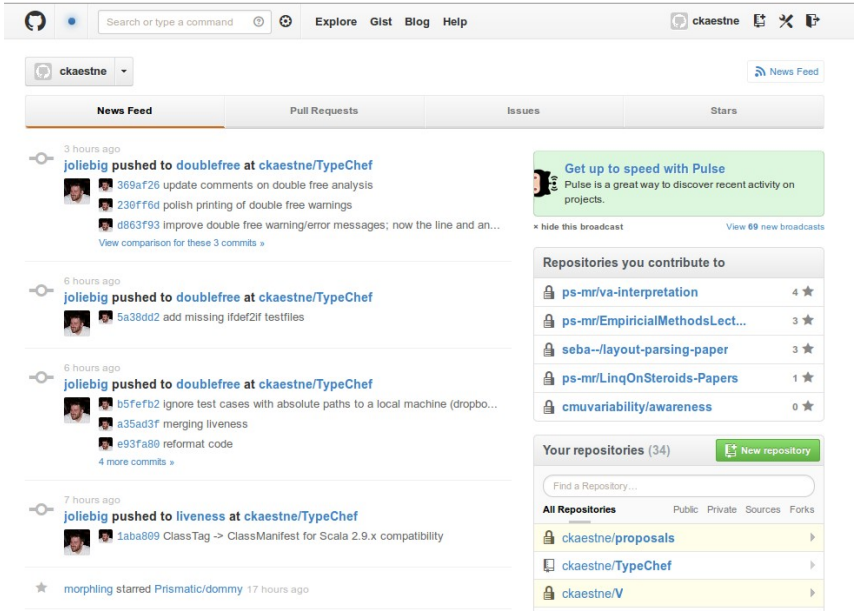
15-313: Foundations of Software Engineering

- Advanced Topics
  - From Design Pattern to Architectures
  - Dynamic and Static Analysis
  - Inspection, Advanced Testing
  - Git, Continuous Integration
- Human Aspects in Software Engineering
  - Teams
  - Process
  - Requirements
  - Economics, Risks
  - Empirical Results
- Concurrency and Security as Themes

Summer Project: Analyzing Github

- Setup Analysis Infrastructure
- Metrics for Active Developers
- Speculative Unit Test Execution

- Unix, Databases +



Summary

- Software product lines: Systematic reuse by planning variations
- From runtime parameters and conditional compilation to design patterns and frameworks
- Analysis for product lines