

ACCU 2009: Stop the Software Architecture Erosion

Bernhard Merkle
Central Unit Research & Development
Software-Engineering
SICK-AG Waldkirch

mailto:Bernhard.Merkle@sick.de



Wide product and technology range



: Industrial Sensors



: Advanced Industrial Sensors



: Encoder



: Industrial Safety Systems



: Auto Ident



: Analyzers & Process Instrumentation



DARPA: Urban/Grand Challenge, Google SV **SICK**



Traffic and Luggage Control



Levels of Static Analysis

- Code, Design, Architectural
- Examples

Architectural Analysis

- Use Cases
- Tool Support,
- Examples
- Pros/Cons

Summary



Possible levels of Static Analysis:



Goal: find, avoid Problems, Increase QA (and measure it)

Micro-Level

- Code
- e.g: =, ==, { },

Macro-Level

- Class-Design
- e.g: by reference, String concat, Exception-Handling

Architecture-Level:

- Layers, Graphs, Subsystems, Compoments, Interfaces
- e.g: Coupling, Dependency, etc...

Tool-support for each level...

Language itself

- Appendix F/ANSI or G/ISO
 - Unspecified behaviour
 - Undefined behaviour
 - Implementation-defined behaviour
 - Locale-specific behaviour

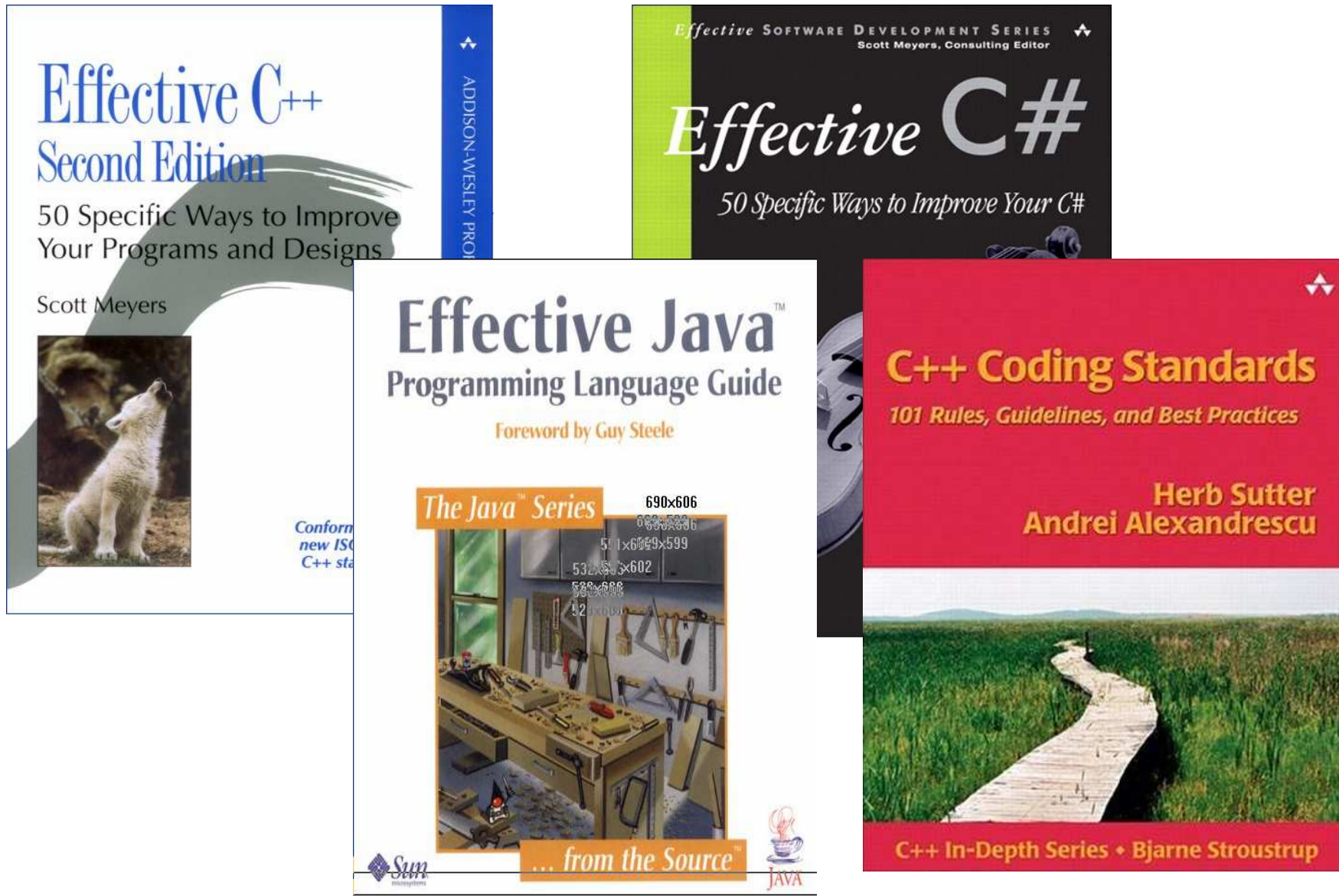
Usage of Language

- Misplacement
- Omission and Addition
- Unexpected behaviour
- Complexity



Macro-Level: Design + Coding Guidelines

SICK



Tool Example Java: findbugs

The screenshot shows the FindBugs application window for an unnamed project. The interface includes a menu bar (Datei, Ansicht, Einstellungen, Hilfe) and a tabbed view with options: 'Nach Klasse', 'Nach Paket', 'Nach Fehlertyp', and 'Übersicht'. A list of bugs is displayed, with 'MS: Mutable static field (70)' selected. Below the list, there are tabs for 'Einzelheiten', 'Quelltext', and 'Anmerkungen'. The 'Einzelheiten' tab is active, showing the title 'Field should be package protected' and a description: 'A mutable static field could be changed by malicious code or by accident. The field could be made package protected to avoid this vulnerability.' The status bar at the bottom left shows the URL 'http://findbugs.sourceforge.net/'.

FindBugs - <<unnamed project>>

Datei Ansicht Einstellungen Hilfe

Nach Klasse Nach Paket Nach Fehlertyp Übersicht

- Eq: Covariant equals() (1)
- HE: Equal objects must have equal hashcodes (4)
- IS2: Inconsistent synchronization (15)
- LI: Unsynchronized Lazy Initialization (1)
- MF: Masked Field (7)
- MS: Mutable static field (70)**
- NN: Naked notify in method (8)
- NP: Null pointer dereference (1)
- RC: Suspicious reference comparison (1)
- SBSC: String concatenation in loop using + operator (10)
- SIC: Inner class could be made static (23)
- Se: Incorrect definition of Serializable class (12)
- SnVI: Serializable class with no Version ID (35)
- UCF: Unchecked control flow (4)

Einzelheiten Quelltext Anmerkungen

Field should be package protected

A mutable static field could be changed by malicious code or by accident. The field could be made package protected to avoid this vulnerability.

FindBugs - <http://findbugs.sourceforge.net/>



IEEE 1471-2000:

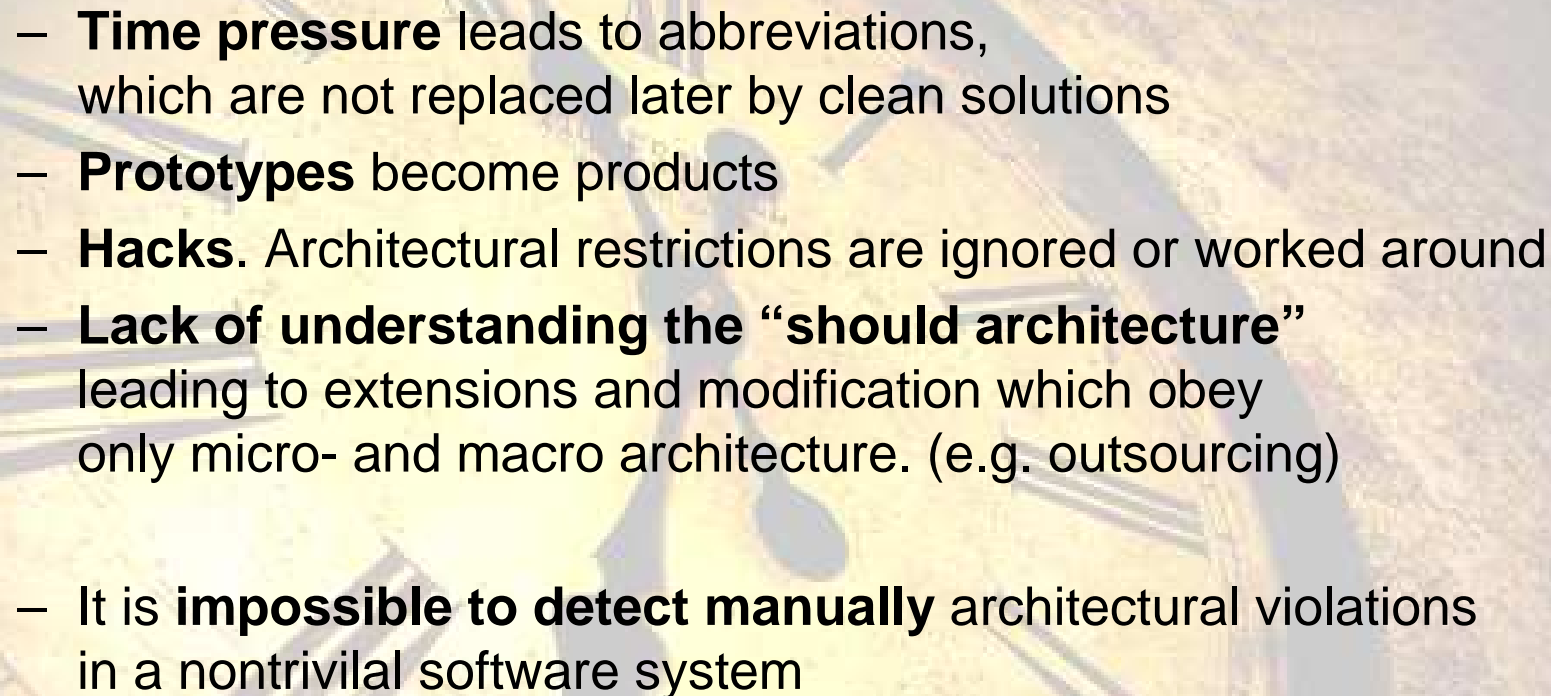
- The fundamental **organization** of a system,
- embodied in its **components**,
- their **relationship** to each other and the environment,
- and the **principles** governing its **design** and **evolution**.

Kruchten: captured in two documents:

- *Software Architecture Document*
- *Software Design Guidelines*

Architecture Erosion ALWAYS happens



- 
- The background of the slide features a faded, high-angle photograph of a large, ornate clock face. The clock has Roman numerals and a prominent shadow cast by the hands, suggesting a late afternoon or early evening setting. The overall tone is warm and slightly desaturated.
- **Time pressure** leads to abbreviations, which are not replaced later by clean solutions
 - **Prototypes** become products
 - **Hacks.** Architectural restrictions are ignored or worked around
 - **Lack of understanding the “should architecture”** leading to extensions and modification which obey only micro- and macro architecture. (e.g. outsourcing)
 - It is **impossible to detect manually** architectural violations in a nontrivial software system

“Sometimes the developers manage to maintain this purity of design through the initial development and into the first release.

More often something goes wrong. The software starts to rot like a piece of bad meat”.

Uncle Bob: “Agile Software Development”

STATIC Analysis Approach

- well known (lint like)
- Reflection, LoadLibrary

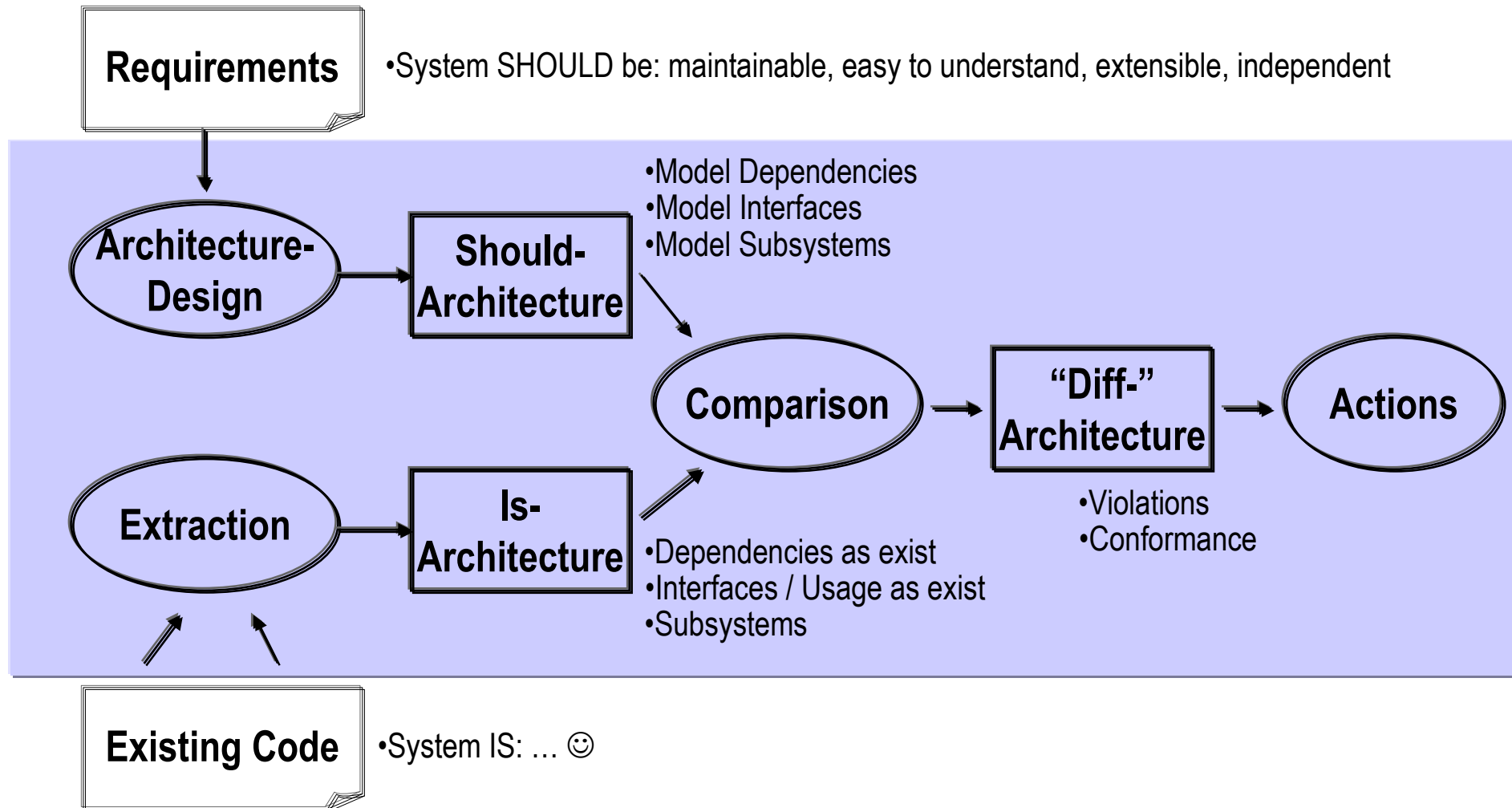
With Tool support

- Pro:
 - automatic, consistent,
 - rule enforcement
 - Bad smell detection
 - Metrics to measure
- Cons:
 - no external Quality
 - Amount of messages



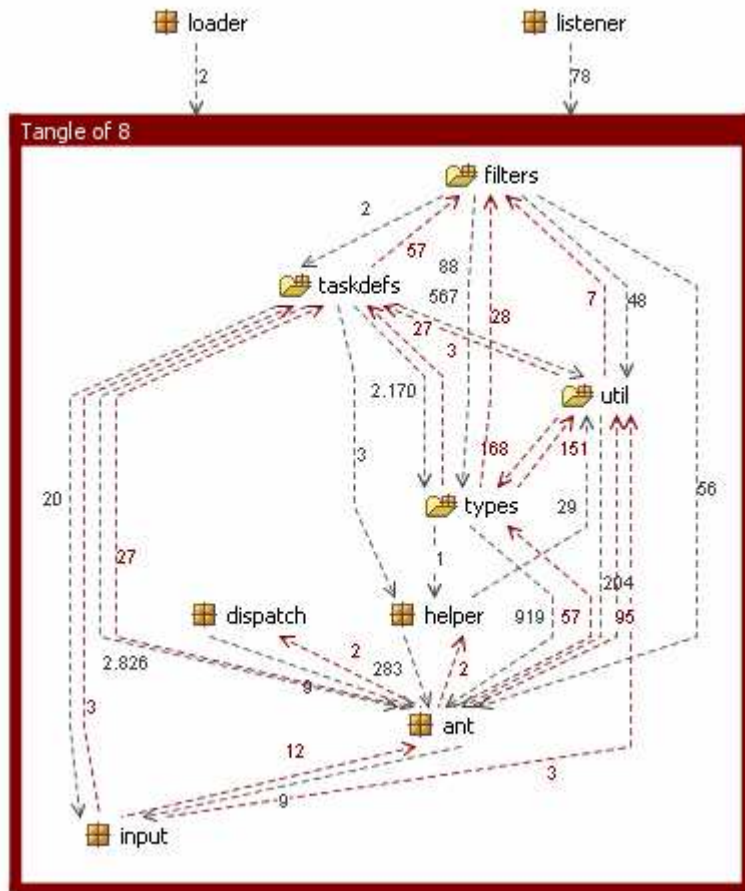
- The Pro is much stronger compared with Code-Lints !!!

Consistency-Analysis



Displaying results:

Graphical and/or numerical

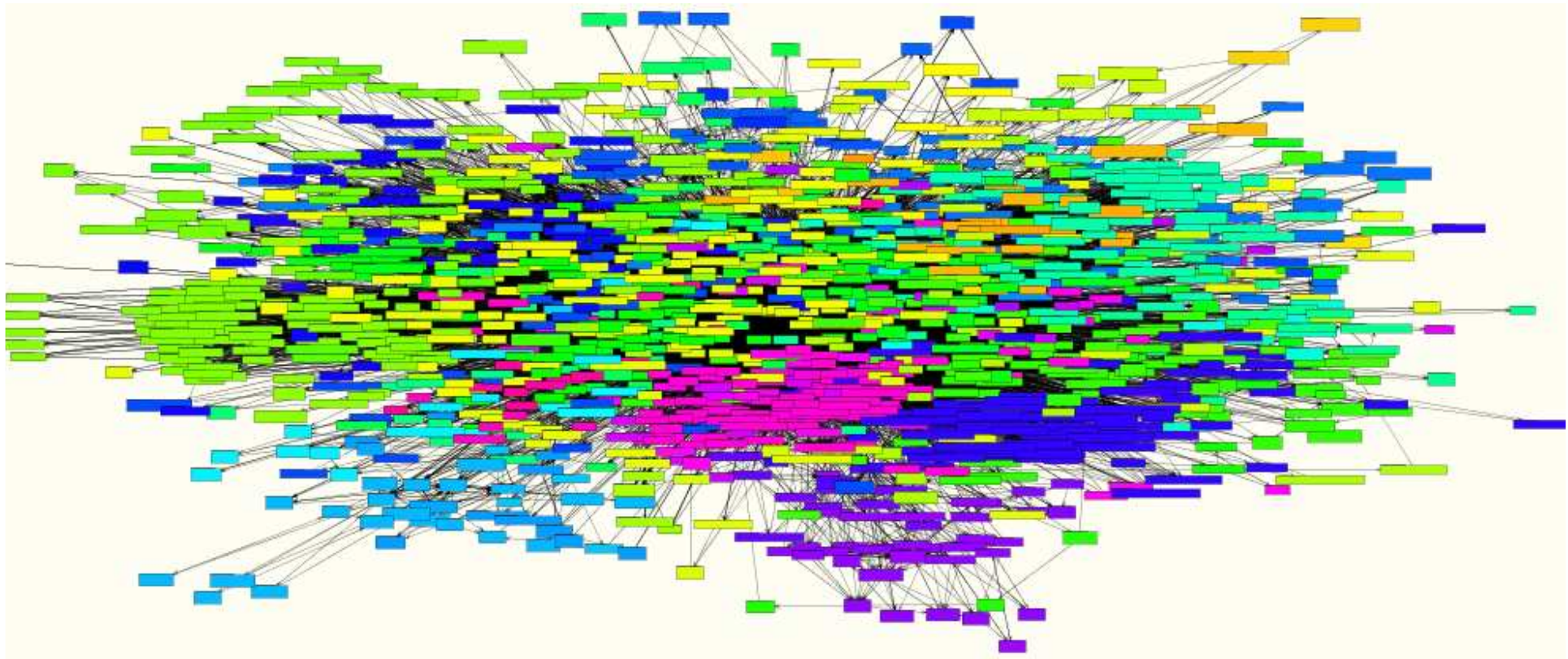


Dependency graph: org.apache.tools.ant

		Tangle of 8									
		loader	listener	dispatch	filters	taskdefs	types	helper	util	ant	input
loader											
listener											
dispatch										2	
filters						57	28		7		
taskdefs			21		2		27		3	27	3
types					88	2...			168	57	
helper						3	1			2	
util			5		48	567	151	29		95	3
ant		2	52		9	56	2...	919	283	204	12
input							20			9	

Rating of Architecture

- JDK 1.5:... 1315 classes in 229 packages all depend on each other !!!
- classes.zip, rt.jar (BIG BALL OF ... ;-)



Still managable ?

SICK



Level Subsystem, Package, File, Class, Operation etc.

- New artefacts
- New dependencies
- New Architecture violations

Early, betimes correction of viloations

Monitoring

- Trendreports
- “outsourcing” projects

Products:

- Sotograph
- Bauhaus
- Structure101

- SonarJ
- Lattix
- Klocwork K7

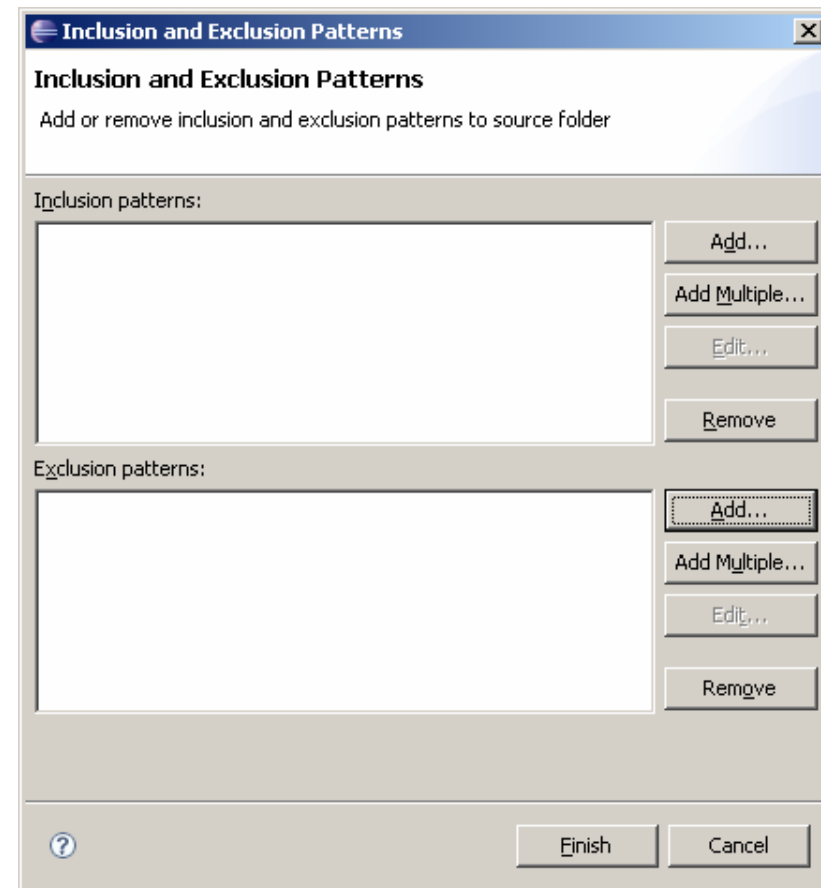
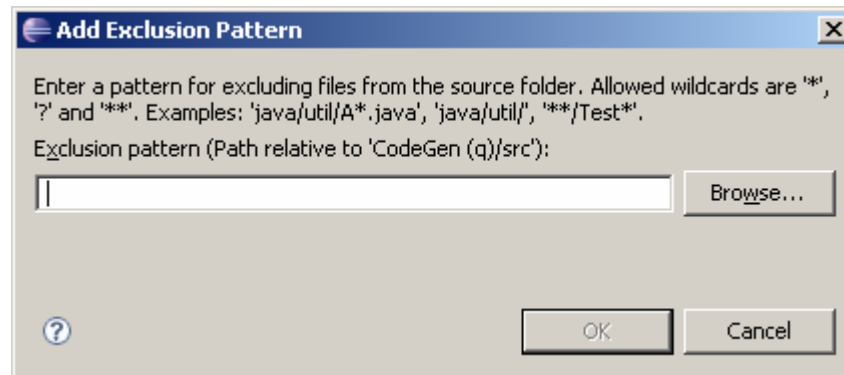


- Others: Semmlecode, CodeCrawler, SeeSoft, XRadars,...

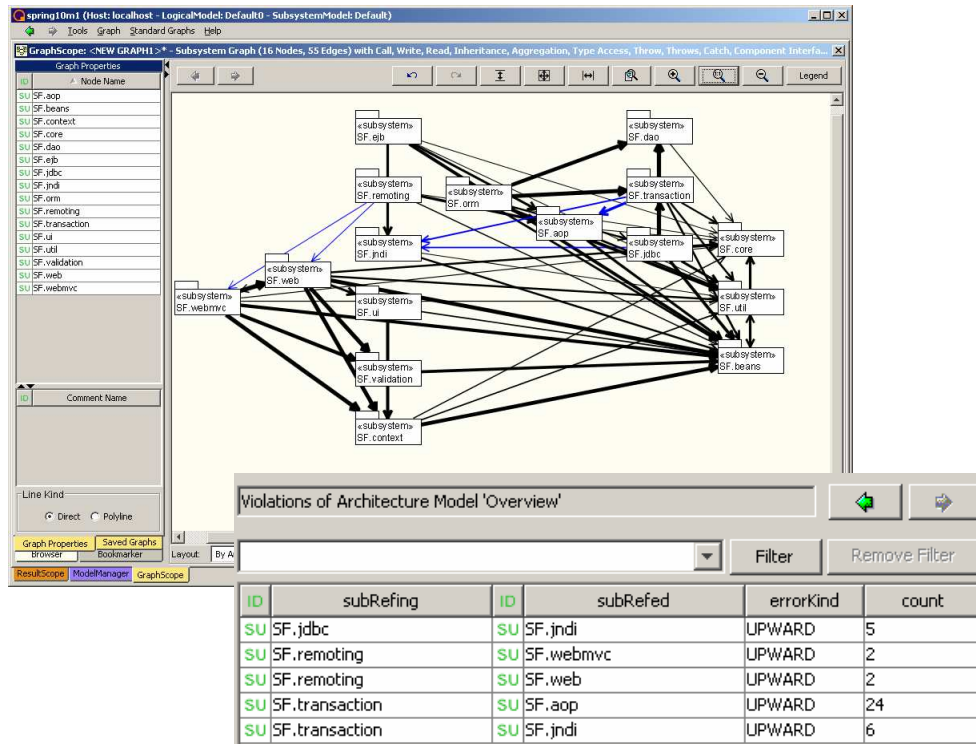
Podcast: <http://se-radio.net/podcast/2008-10/episode-115-architecture-analysis>

Basic approaches

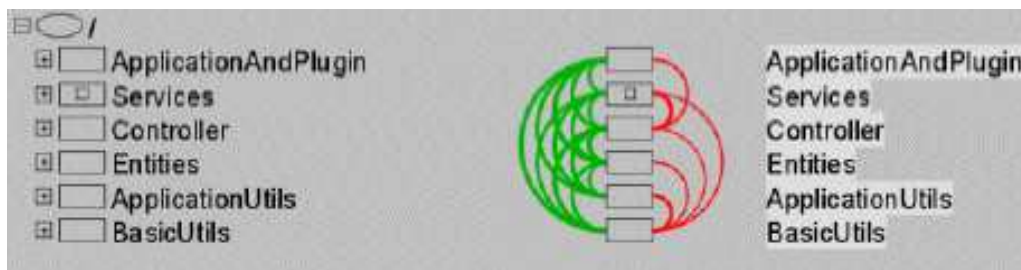
- Your makesystem...
- makedepend, jdepend
- RE code into UML model
- Eclipse (Java Build Path)



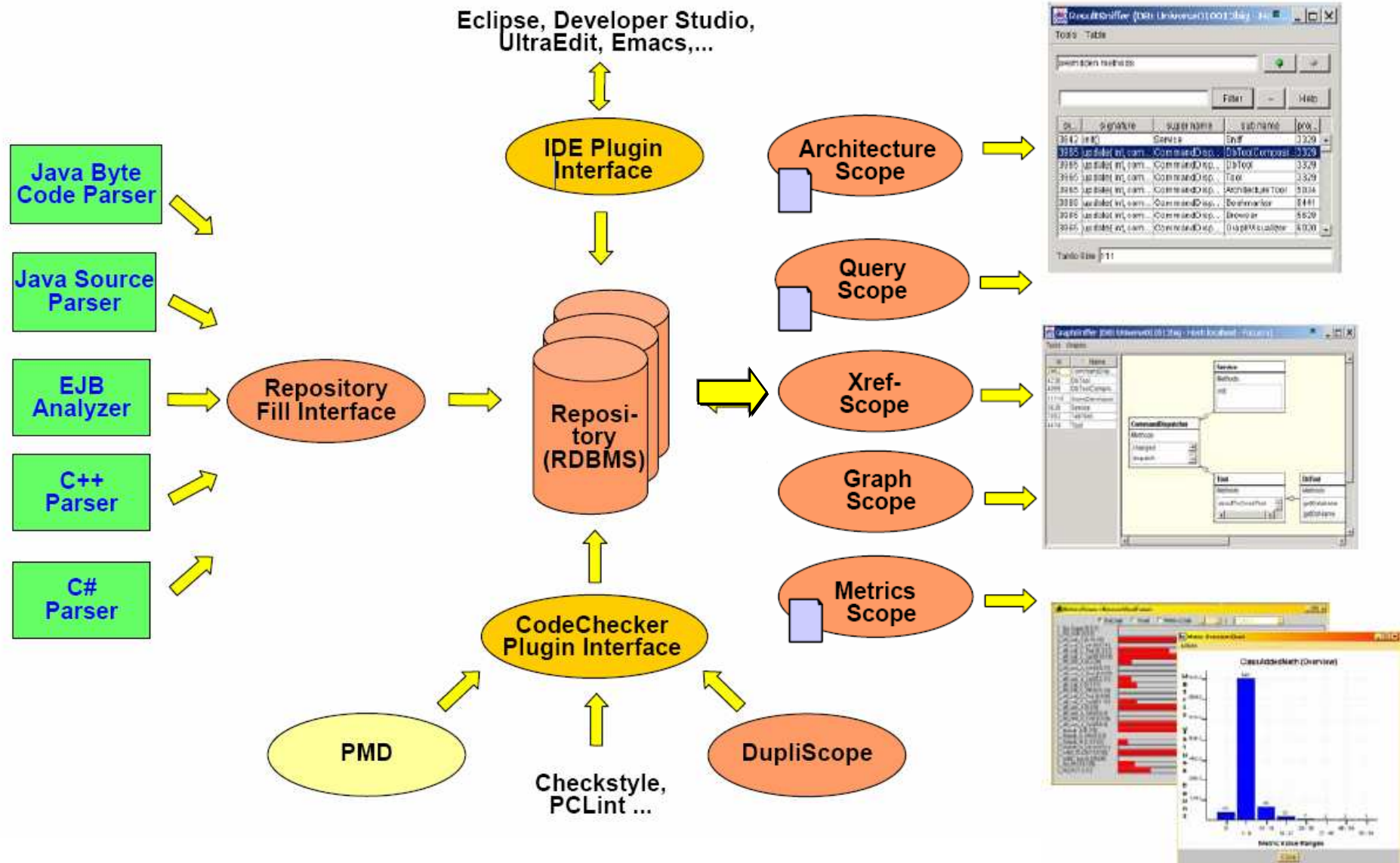
Sotograph: Overview



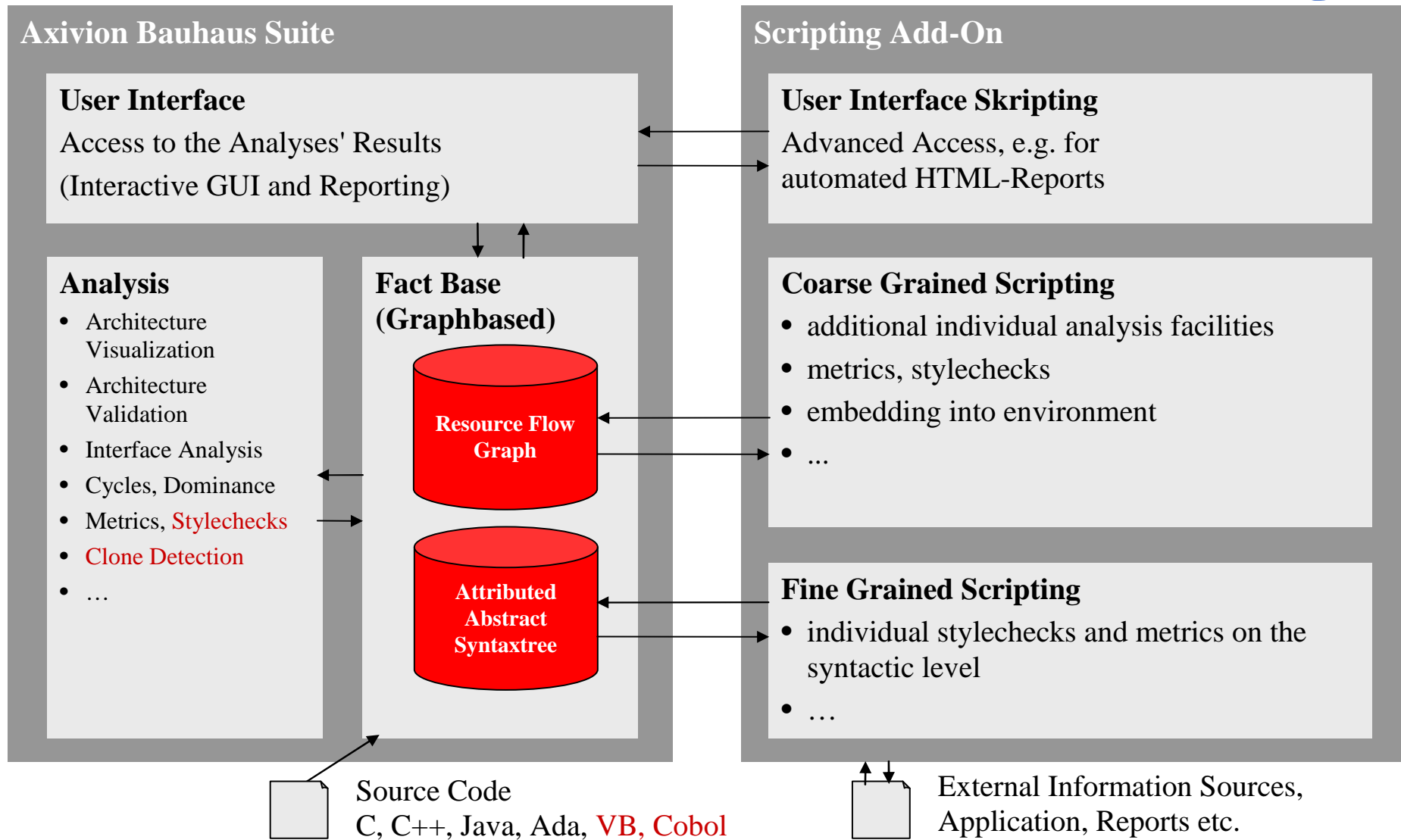
- VERY powerfull
- Infos via Table + Graph
- Strong layout algorithms
- Known since 2003
- Mysql DB, open schema
- GUI Client, Web Report
- About 200+ Metrics
- Arbitrary User queries
- Trend Analysis
- Virtual Refactoring
- Java, C++, C#, source parser

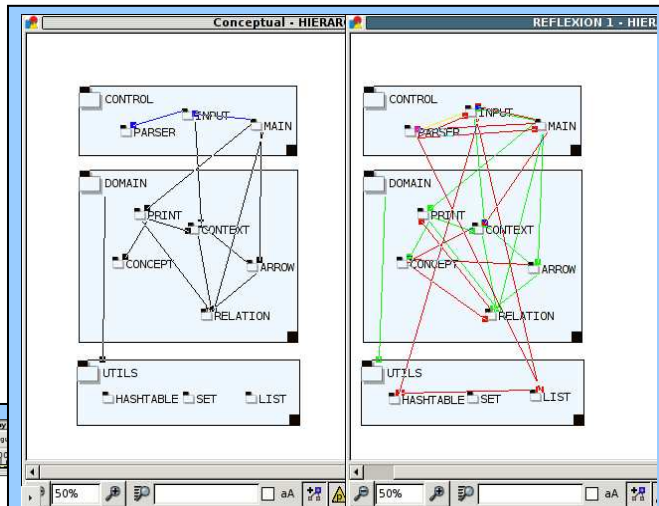


- SotoArch since 2007



Axivion Bauhaus Suite



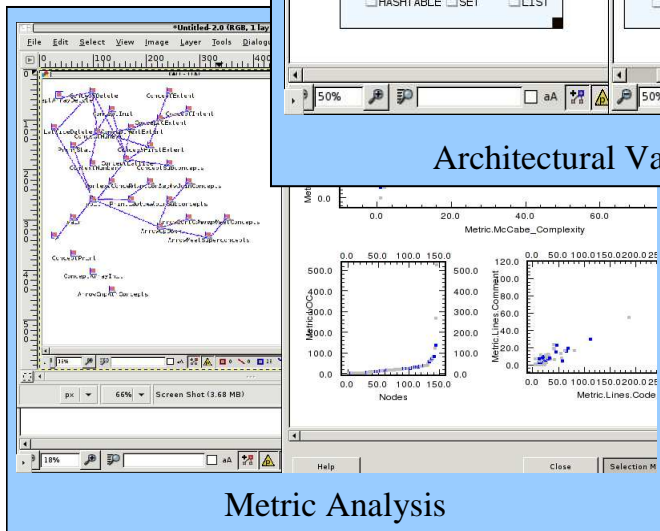


Architectural Validation

```
home/eisenbits/bauhaus/bauhaus-tools/example/lib/lib.c
ListEntry *prev ;
if (entry == entry->list->first) {
    /* first entry in chain */
    entry->list->first = entry->next ;
} else {
    for (prev = entry->list->first ; prev = prev->next ;
        if (prev == LIST_NULL) {
            panic ("malformed List in %s:%i",
                FILE , LINE ) ;
        }
        if (prev->next == entry) {
            /* found - remove from chain */
            prev->next = entry->next ;
            break ;
        }
}

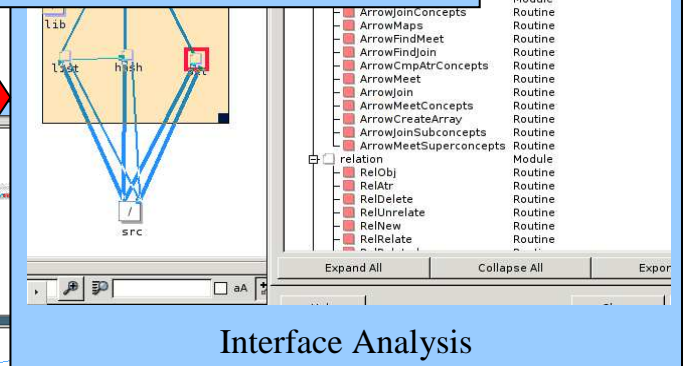
home/eisenbits/bauhaus/bauhaus-tools/example/lib/hash.c
HashEntry *prev ;
if (*entry->bucket == entry) {
    /* first entry in bucket */
    *entry->bucket = entry->next ;
} else {
    /* search through chain */
    for (prev = *entry->bucket ; prev = prev->next ;
        if (prev == HASH_NULL) {
            panic ("malformed chain in %s:%i",
                FILE , LINE ) ;
        }
        if (prev->next == entry) {
            prev->next = entry->next ;
            break ;
        }
}
/* decrease number of entries
```

Clone Detection

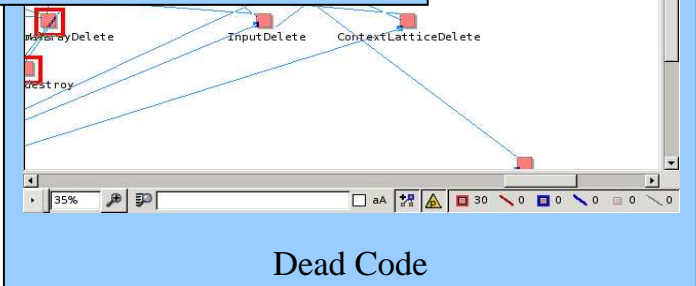


Metric Analysis

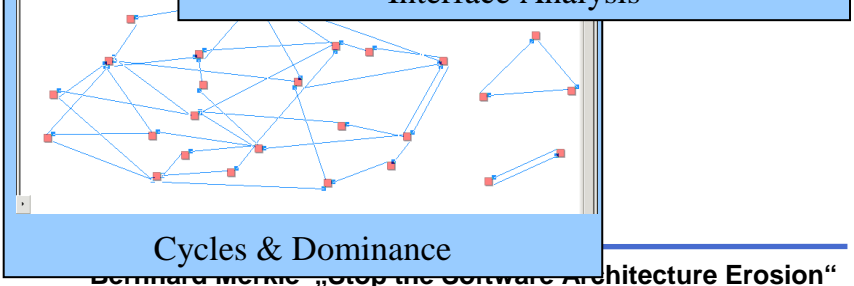
Axivion Bauhaus Suite



Interface Analysis



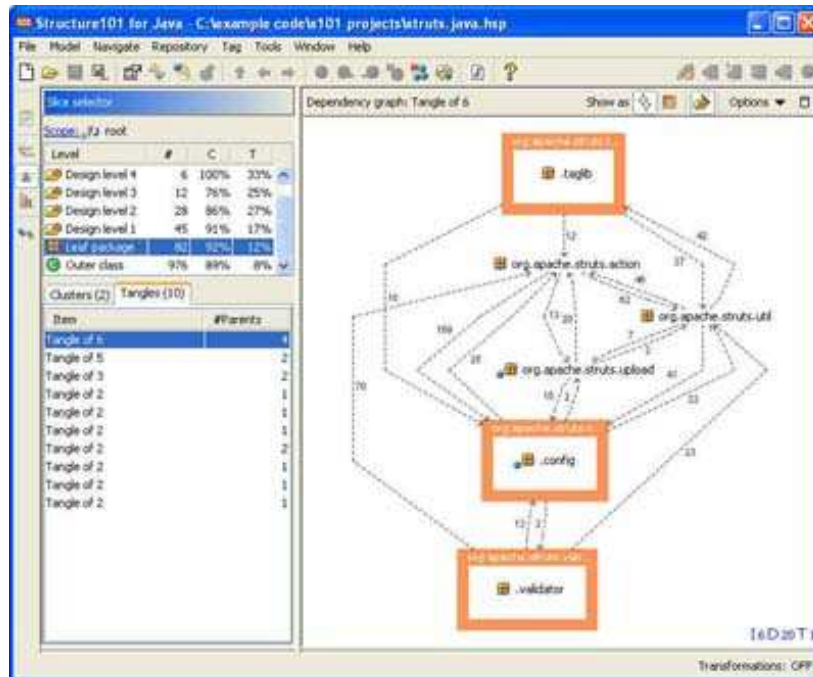
Dead Code



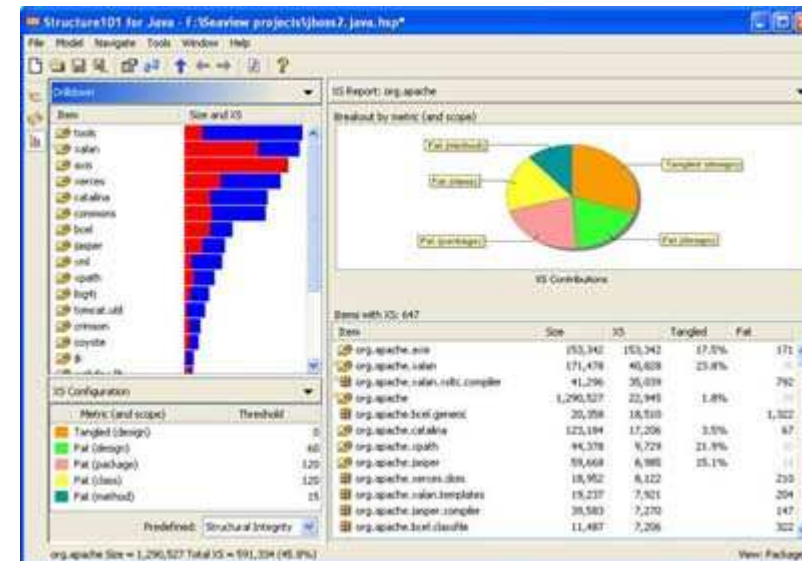
Cycles & Dominance

Structure101: Overview

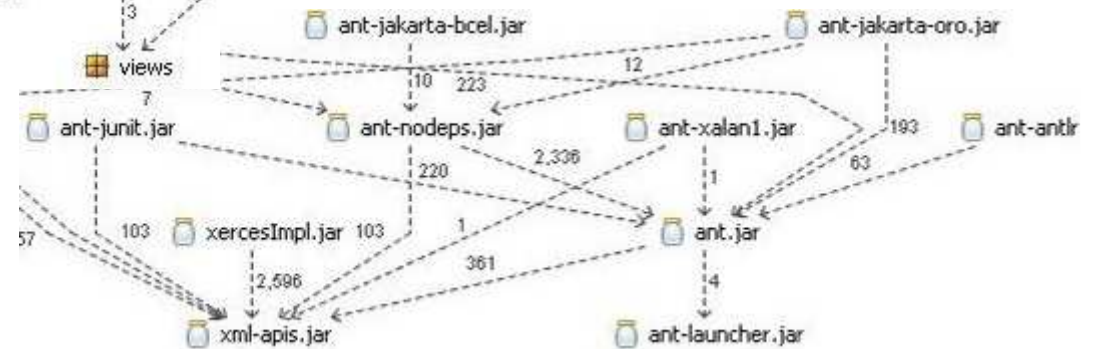
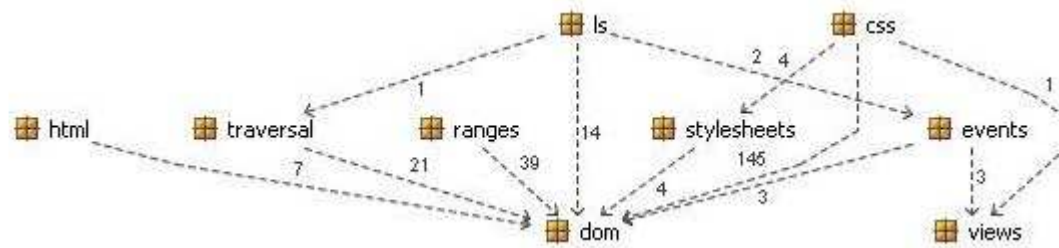
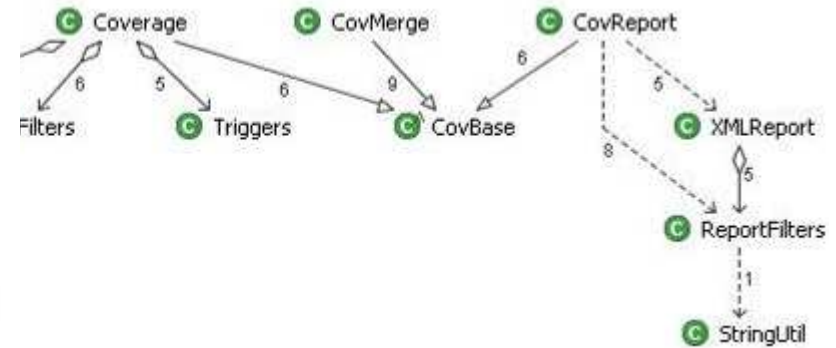
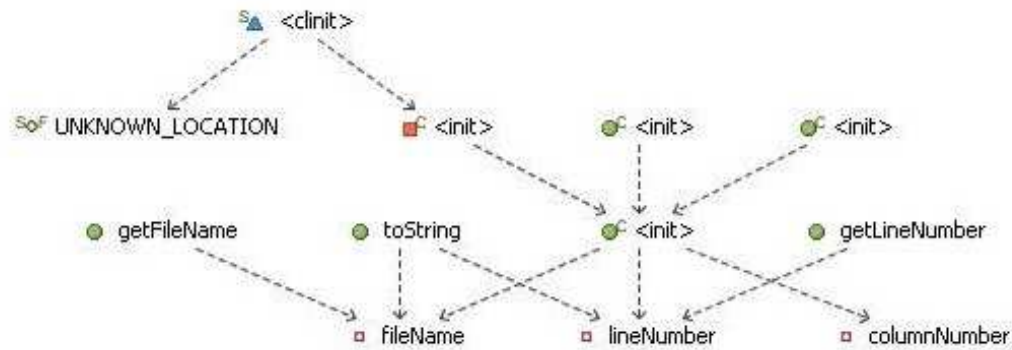
SICK



- Java, C++, Ada
- Generic version
- Infos via DSM + Graphs
- Known since 2005
- Repository/DB server
- Fat-Client, Web
- Lightweight approach



Structure101: Dependency Management



Build/analysis time

Integration in IDE, own GUI

Continuous build, commandline,

Analysis results, drill down

Granularity (dependency types)

Modeling styles (API, regexp, Graphical)

Usage by developer, architect

Report, Trenddata, Web-Export

Complexity, Price, Setup

Dependency Types

The screenshot displays the 'Query by Criteria' application interface. On the left, a tree view under 'Edges' lists various dependency types, including 'Source_Dependency', 'Call', 'Inheritance', and 'Reference'. The 'Project properties' dialog box is open, showing the 'Dependency types' tab with options to 'Show all dependencies (default)' or 'Hide the following dependency types'. A 'Specify Displayed Reference ...' dialog box is also open, allowing users to select specific reference types such as 'AGGREGATION', 'CALL', 'CATCH', 'COMPOONENTCALL', 'COMPOONENTIMPLEMENTATION', 'COMPOONENTINTERFACECALL', 'INHERITANCE', 'READ', 'THROW', 'THROWS', 'TYPEACCESS', and 'WRITE'. The 'INHERITANCE' checkbox is checked in this dialog.

Query by Criteria

Nodes | Edges | Parameters

Edges

Types (collapse all)

- Belongs_To
- Comparison
 - Absence
 - Convergence
 - Divergence
- Maps_To
- Source_Dependency**
 - Based_On_Type
 - Type_Synonym_To
- Call
 - Dispatching_Call
 - Dynamic_Call
 - Static_Call
- Clones
- Communication
- Declared_In
- Friend_Of
- Inheritance
 - Extends
 - Member_Inheritance
 - Inherits
 - Overrides
- Instantiates
- Internal_Access
- Local_Var_Of_Type
- Of_Type
- Primitive_For
- Reference
 - Address
 - Member_Address
 - Routine_Address
 - Method_Address
 - Variable_Address
 - Set
 - Member_Set
 - Variable_Set
 - Use
 - Member_Use
 - Variable_Use
- Signature
 - Parameter_Of_Type
 - Return_Type
 - Template_Parameter_Of
- Throws_Exception

Project properties

Main settings | Pre-transformations | Excludes | **Dependency types** | File extensions

Typically you will want to see all dependencies in the code-base.

However, for some use cases, you may wish to hide some dependencies to obtain a (filtered) dependency model.

Show all dependencies (default)

Hide the following dependency types:

<input type="checkbox"/> calls	<input type="checkbox"/> has	<input type="checkbox"/> has param
<input type="checkbox"/> includes	<input type="checkbox"/> is a	<input type="checkbox"/> references
<input type="checkbox"/> returns	<input type="checkbox"/> specializes	<input type="checkbox"/> subclasses

Select all | Deselect all

OK | Cancel

Specify Displayed Reference ...

select the reference types Specify Displayed

<input type="checkbox"/> AGGREGATION
<input type="checkbox"/> CALL
<input type="checkbox"/> CATCH
<input type="checkbox"/> COMPOONENTCALL
<input type="checkbox"/> COMPOONENTIMPLEMENTATION
<input type="checkbox"/> COMPOONENTINTERFACECALL
<input checked="" type="checkbox"/> INHERITANCE
<input type="checkbox"/> READ
<input type="checkbox"/> THROW
<input type="checkbox"/> THROWS
<input type="checkbox"/> TYPEACCESS
<input type="checkbox"/> WRITE

Select All | Select None

Ok | Cancel

Modeling Approach: SotoArc

The screenshot shows the SotoArc M3.5 software interface. The title bar reads "Sotoarc M3.5 - findbugs20090416_121 @ localhost - findbugs_121 - Advanced Mode". The menu bar includes "Sotoarc", "Projects", "Model Set", "Options", "Extras", and "Help".

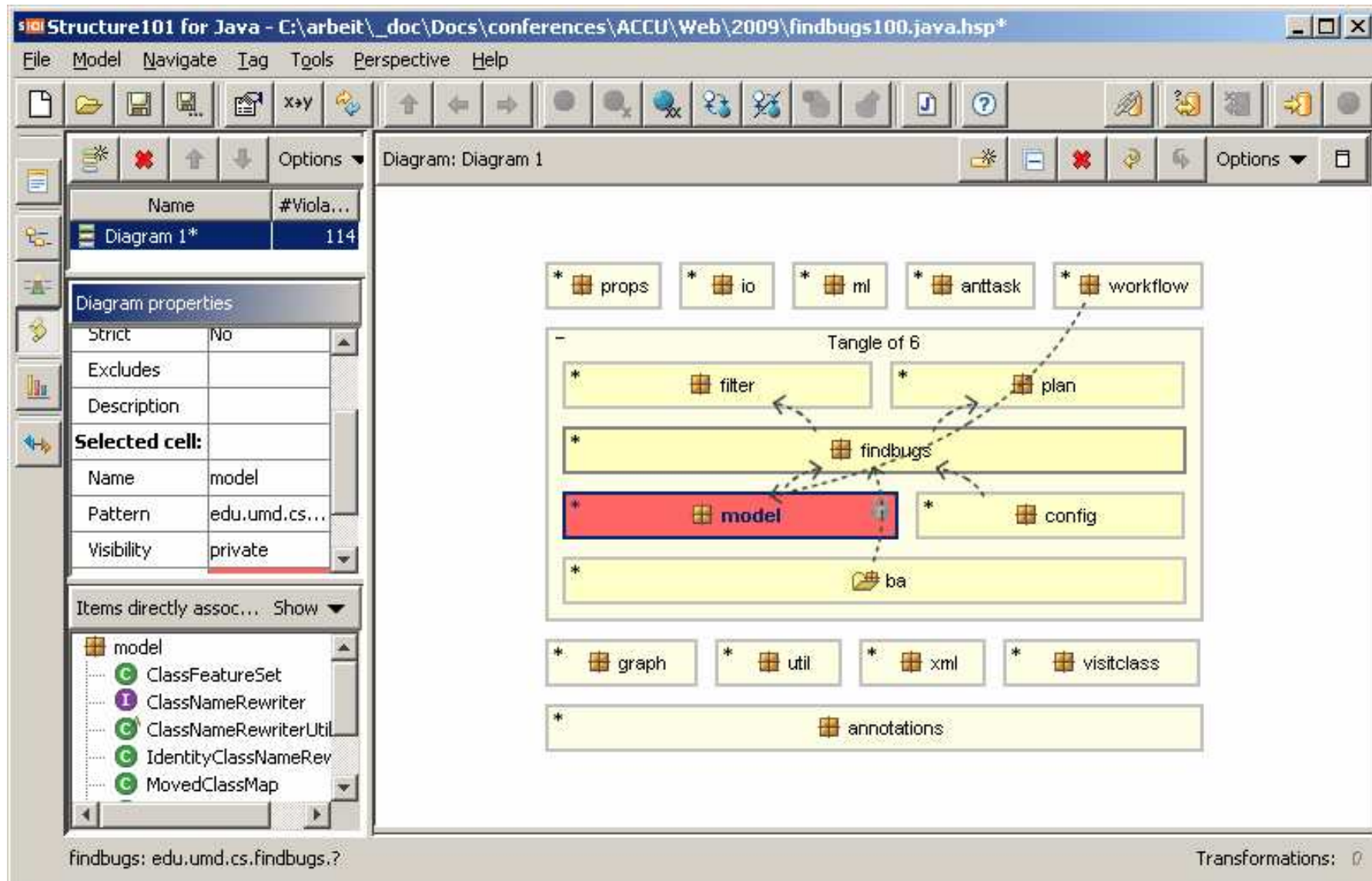
On the left, there is a sidebar with tabs for "Architecture", "Restructured Code", "Cycles", "Metrics", "Original Code", and "Models". Below these is a tree view showing "LM Base (pckgs: Java packages)".

Below the sidebar is a table titled "Operations of Selected Model":

...	Description
L	M	✓	Filter nodes (just hide) 'j...
L	M	✓	Create unrestricted subs...
L	M	✓	Create 'fb.findbugs' (unr...
L	M	✓	Create 'util' (layer) bene...
L	M	✓	Create 'io-ba' (layer) ben...
L	M	✓	Create 'filter-classfile-co...
L	M	✓	Create 'findbugs' (layer) ...
L	M	✓	Create 'model' (layer) be...
L	M	✓	Create 'gui' (layer) bene...
L	M	✓	Create 'top' (layer) bene...
L	M	✓	Fix child order of 'j'

The main workspace shows a "Architecture Modeling View" with a tree on the left and a network diagram on the right. The tree lists packages: top, gui, model, findbugs, filter-classfile-config, io-ba, util, and PACKAGES (Java packages). The network diagram shows a complex web of green and red connections between nodes.

Modeling Approach: Structure101



Modeling Approach: Bauhaus

Architectural Mapping

State	Architectural View "ARCHITECTURE" (expand all)	Mappe	State	Hierarchy View "MODULE" (expand all)	Mapped To
<input checked="" type="checkbox"/>	CONTROL		<input checked="" type="checkbox"/>	src	
<input checked="" type="checkbox"/>	INPUT		<input checked="" type="checkbox"/>	arrow	ARROW
<input checked="" type="checkbox"/>	MAIN		<input checked="" type="checkbox"/>	concept	CONCEPT
<input checked="" type="checkbox"/>	PARSER		<input checked="" type="checkbox"/>	context	CONTEXT
<input checked="" type="checkbox"/>	ARROW		<input checked="" type="checkbox"/>	input	INPUT
<input checked="" type="checkbox"/>	CONCEPT		<input checked="" type="checkbox"/>	lex.yy	PARSER
<input checked="" type="checkbox"/>	CONTEXT		<input checked="" type="checkbox"/>	main	MAIN
<input checked="" type="checkbox"/>	PRINT		<input checked="" type="checkbox"/>	panic	
<input checked="" type="checkbox"/>	RELATION		<input checked="" type="checkbox"/>	print	PRINT
<input checked="" type="checkbox"/>	UTILS		<input checked="" type="checkbox"/>	relation	RELATION
<input checked="" type="checkbox"/>	HASHTABLE		<input checked="" type="checkbox"/>	scanner	PARSER
<input checked="" type="checkbox"/>	LIST		<input checked="" type="checkbox"/>	version	
<input checked="" type="checkbox"/>	SET		<input checked="" type="checkbox"/>	y.tab	PARSER

ARCHITECTURE - HIERARCHY

```

graph TD
    CONTROL[CONTROL]
    DOMAIN[DOMAIN]
    UTILS[UTILS]
    CONTROL --- DOMAIN
    CONTROL --- UTILS
  
```

Gravis - A:/EXAMPLE/SRC/CONCEPTS.RFG

State	Role	Hierarchy	Name	□	—	□	—	□	—
<input checked="" type="checkbox"/>			BASE	389	1752	0	0	0	0
<input checked="" type="checkbox"/>			CYCLES CALL	3	3	0	0	0	0
<input checked="" type="checkbox"/>			ENVIRONMENT	240	0	0	0	0	0
<input checked="" type="checkbox"/>			CALL	201	407	0	0	0	0
<input checked="" type="checkbox"/>			MODULE	427	423	0	0	0	0
<input checked="" type="checkbox"/>			LIFTEDMODULE	427	4427	0	0	0	0
<input checked="" type="checkbox"/>			USER	94	87	0	0	0	0
<input checked="" type="checkbox"/>			CLONES	49	37	0	0	0	0
<input checked="" type="checkbox"/>			ARCHITECTURE	14	24	0	0	3	1
<input checked="" type="checkbox"/>			MAPPING	24	13	0	0	0	0
<input checked="" type="checkbox"/>			BASE-ENV	170	1029	0	0	0	0
<input checked="" type="checkbox"/>			CALL-ENV	103	222	0	0	0	0
<input checked="" type="checkbox"/>			MODULE ENV	187	185	0	0	0	0

No source position information available

Examples for Architectural Analysis

SICK

-ant

- easy to use make system for java

-findbugs

- code/design level analysis tool
- Implementation of Effective Java book (Joshua Bloch)

-boost

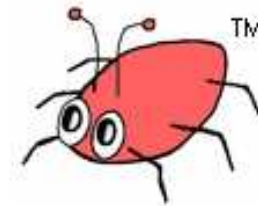
- collection of (meta) template libraries
- close relation to C++(0x) standardization

-poco

- cross platform library (embedded development)

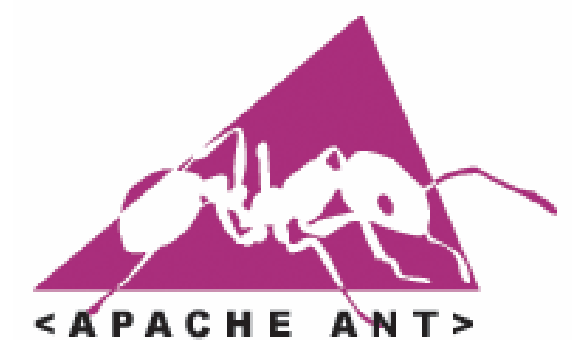
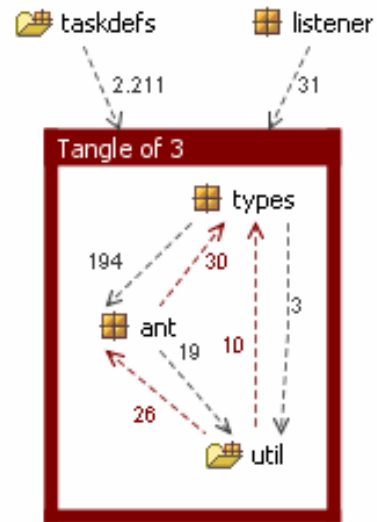
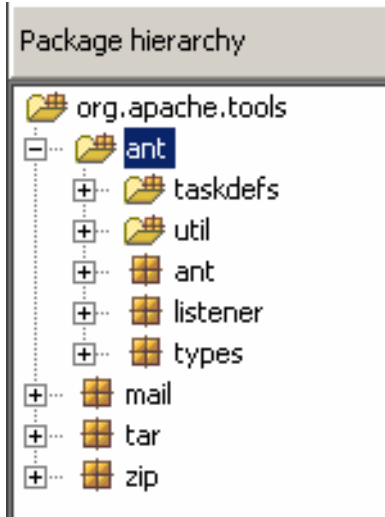
-flightgear

- open source flight simulator

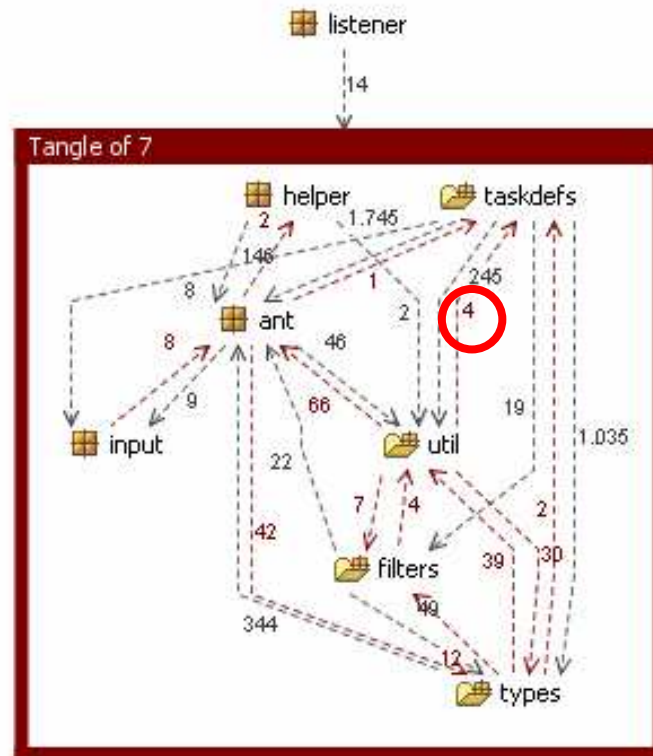
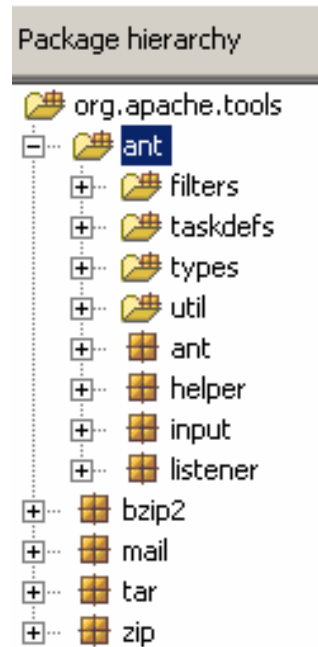


ANT 1.4.1: Architectural Analysis

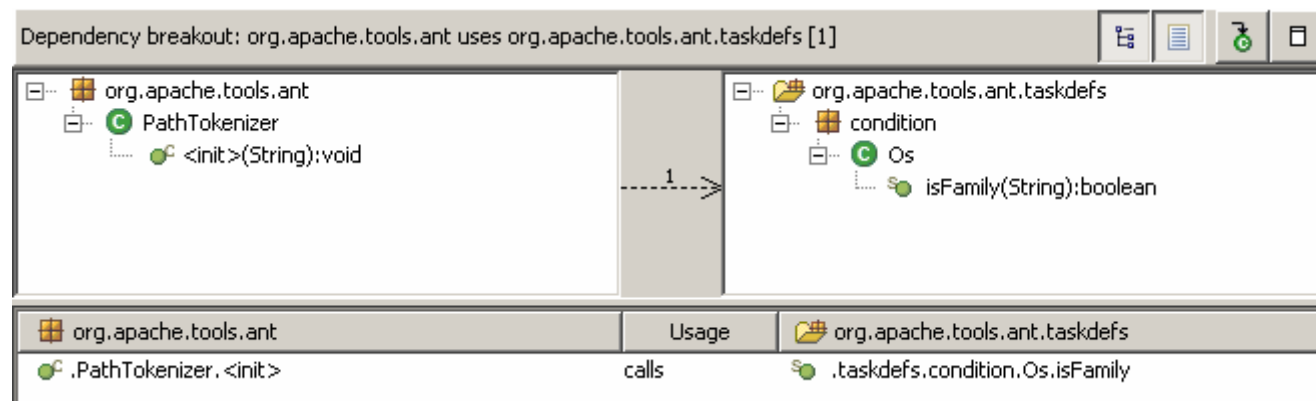
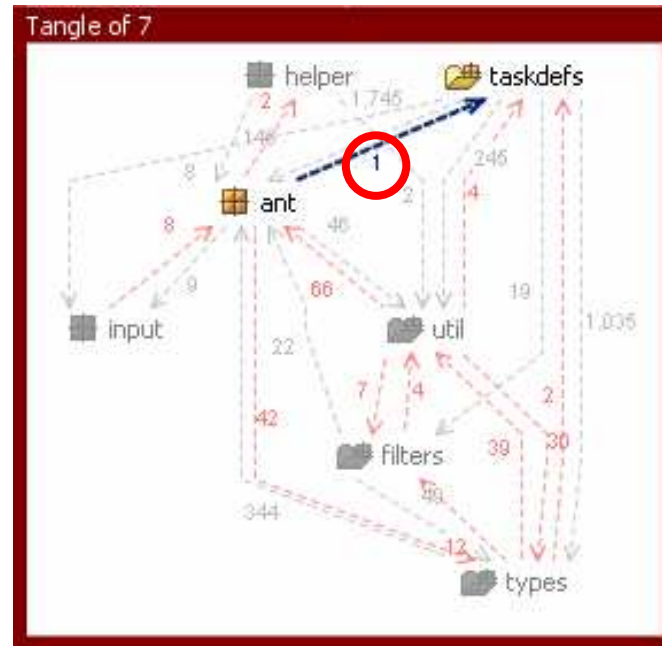
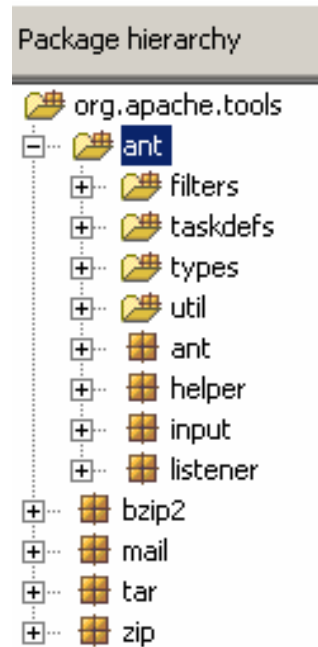
SICK



ANT 1.5.2: Architectural Analysis

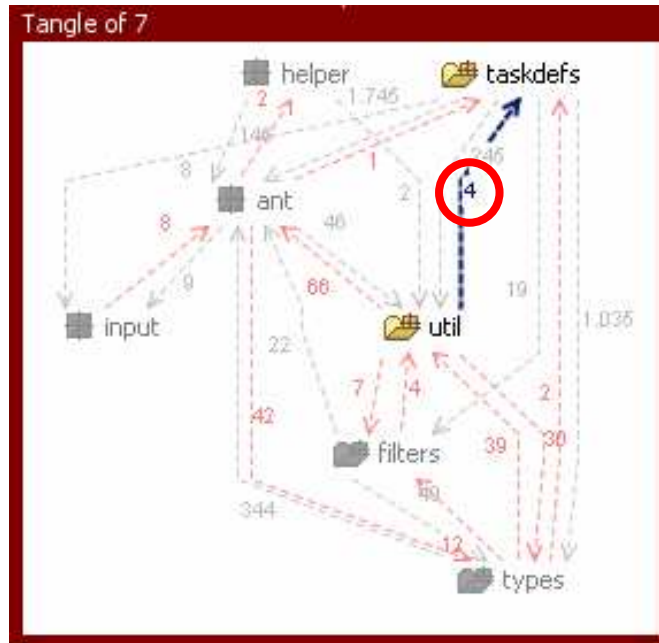
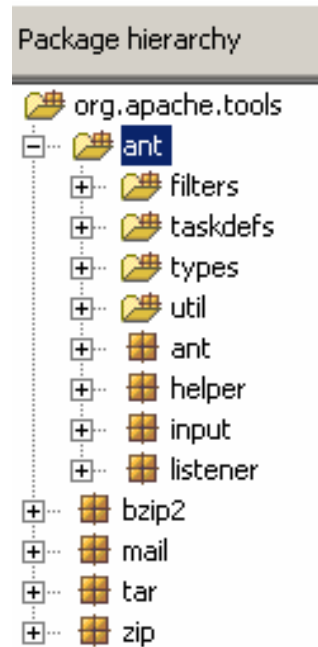


ANT 1.5.2: Architectural Analysis



ANT 1.5.2: Architectural Analysis

SICK

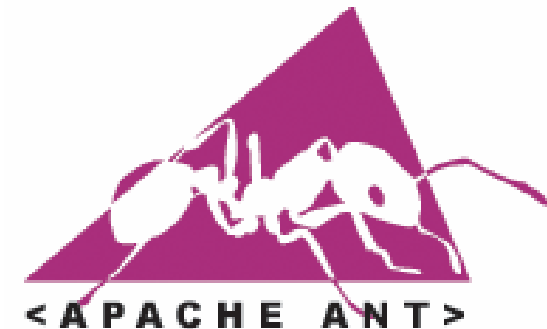
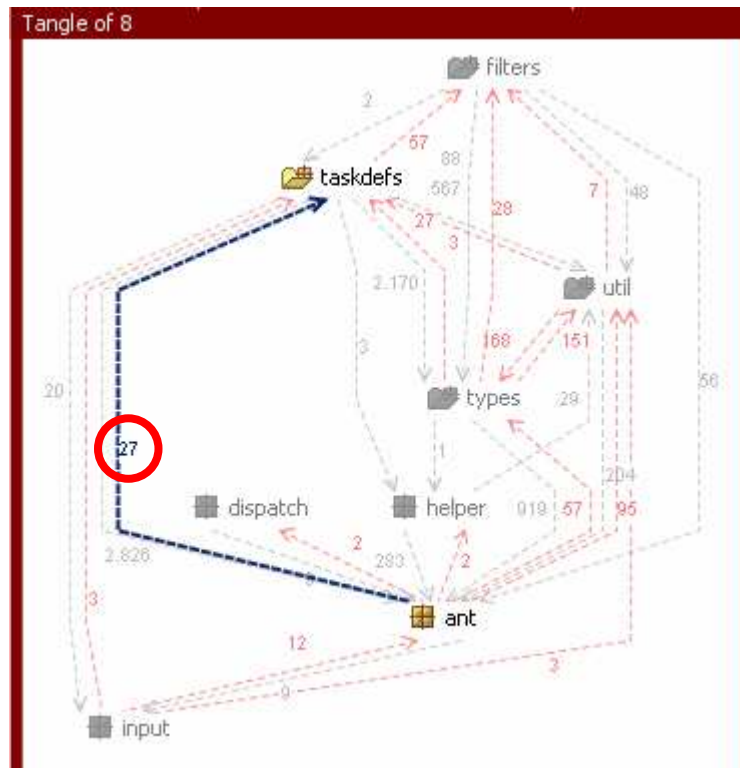
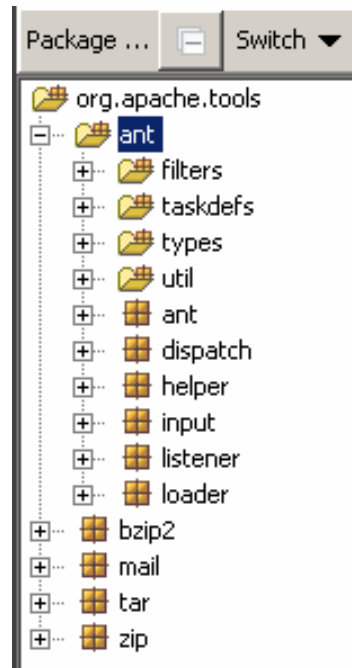


Dependency breakout: org.apache.tools.ant.util uses org.apache.tools.ant.taskdefs [4]

org.apache.tools.ant.util	Usage	org.apache.tools.ant.taskdefs
util.FileUtils.<init>	calls	taskdefs.condition.Os.isFamily
util.JavaEnvUtils.<clinit>	calls	taskdefs.condition.Os.isName
util.JavaEnvUtils.<clinit>	calls	taskdefs.condition.Os.isFamily
util.ResourceUtils.selectOutOfOrderNames	calls	taskdefs.condition.Os.isFamily

ANT 1.7.1: Architectural Analysis

SICK

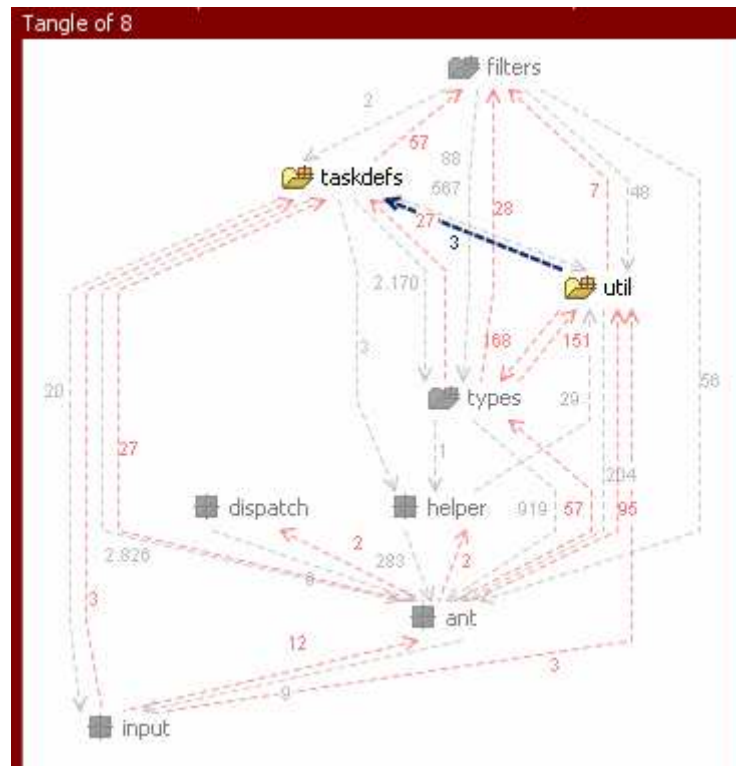
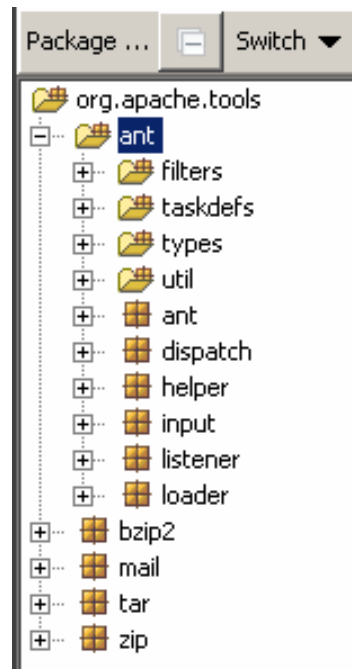


Dependency breakout: org.apache.tools.ant uses org.apache.tools.ant.taskdefs [27]

org.apache.tools.ant	Usage	org.apache.tools.ant.taskdefs
.ComponentHelper.checkNamespace	references	.taskdefs.Typedef
.ComponentHelper.checkNamespace	calls	.taskdefs.Typedef.<init>
.DirectoryScanner.<clinit>	calls	.taskdefs.condition.Os.isFamily
.PathTokenizer.<init>	calls	.taskdefs.condition.Os.isFamily

ANT 1.7.1: Architectural Analysis

SICK



Dependency breakout: org.apache.tools.ant.util uses org.apache.tools.ant.taskdefs [3]

org.apache.tools.ant.util

- util
 - FileUtils
 - JavaEnvUtils

org.apache.tools.ant.taskdefs

- condition
 - Os
 - isFamily(String):boolean
 - isName(String):boolean

org.apache.tools.ant.util	Usage	org.apache.tools.ant.taskdefs
.util.FileUtils.<clinit>	calls	.taskdefs.condition.Os.isFamily
.util.JavaEnvUtils.<clinit>	calls	.taskdefs.condition.Os.isFamily
.util.JavaEnvUtils.<clinit>	calls	.taskdefs.condition.Os.isName

Transformations: 0

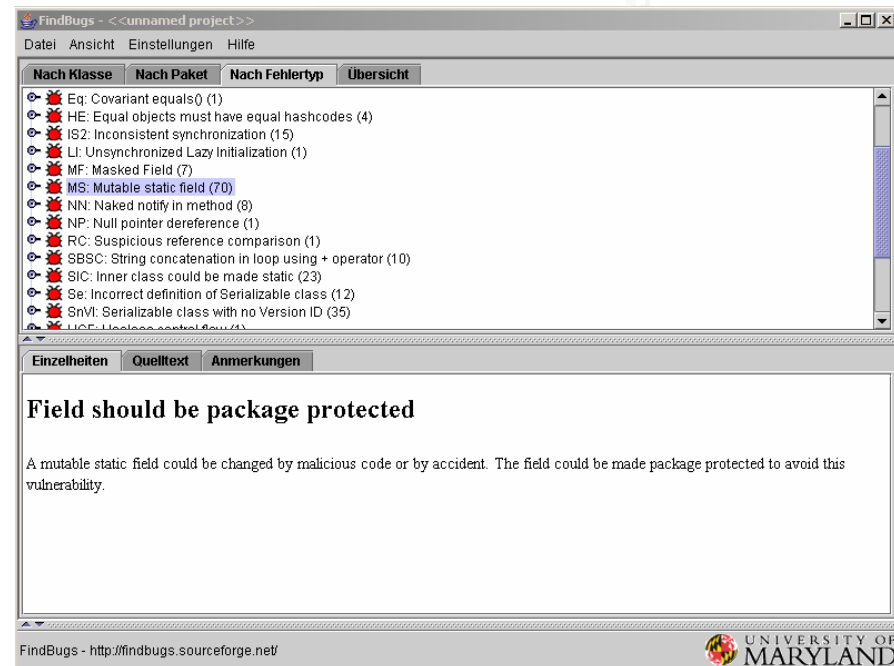
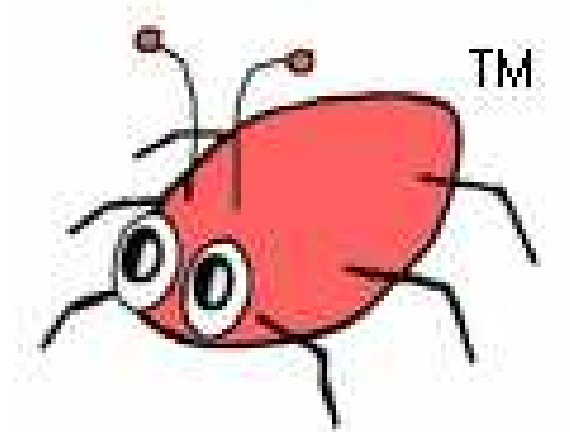
Still manageable ?

SICK

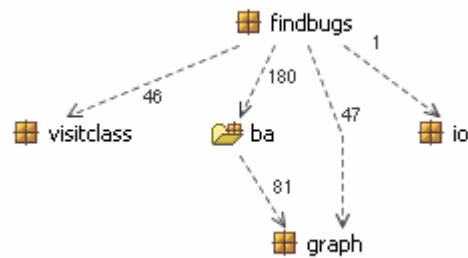


-Findbugs

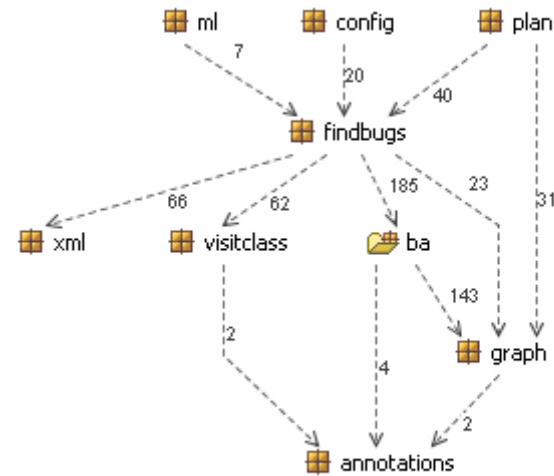
- do codelevel lints care about architecture ?
- Is there a architecture ?
- Is there a erosion ?
- Do AA-Tool work well ?



Findbugs: the first years

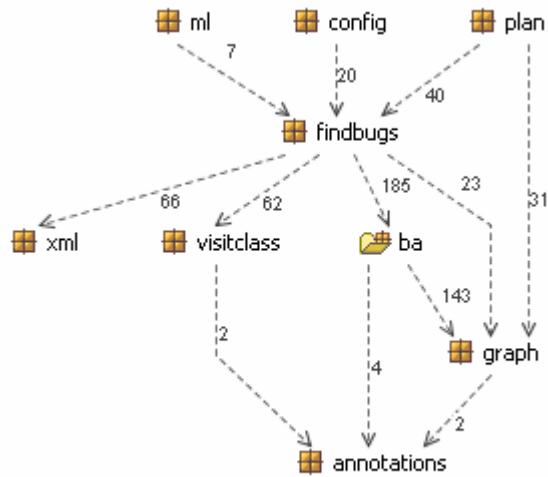


0.7.2 (03/2004)

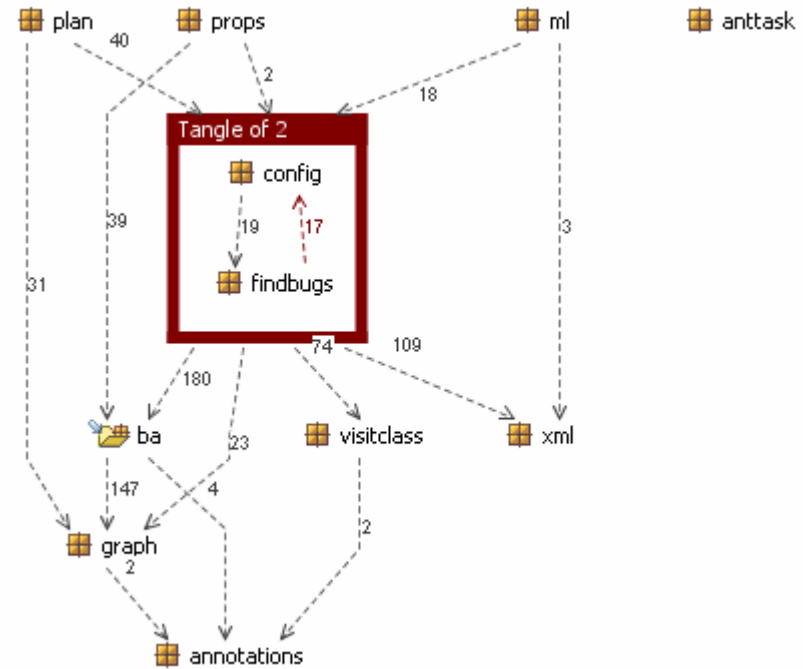


0.8.6 (10/2004)

Findbugs: the first years

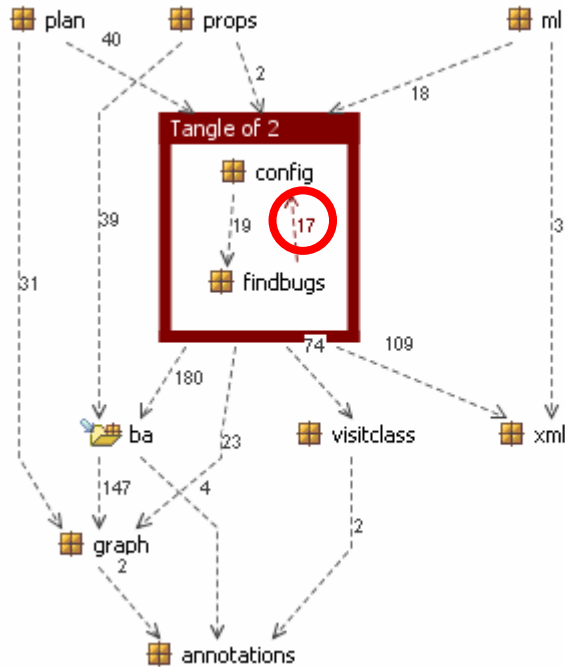


0.8.6 (10/2004)



0.8.7 (05/2005)

Findbugs 0.8.7: Architectural Analysis



Log Messages - <http://findbugs.googlecode.com/svn>

From: 01.03.2005 To: 01.05.2005 Messages, authors and paths

Revision	Actions	Author	Date	Message
3716		daveho	22:20:33, Sonntag, 6. März 2005	Eliminated some inadvertent autoboxing.
3715		daveho	22:11:19, Sonntag, 6. März 2005	Added getUserPreferencesFile(). Modified to use IO.writeF
3714		daveho	22:10:25, Sonntag, 6. März 2005	Initial checkin
3713		daveho	21:42:31, Sonntag, 6. März 2005	Fixed copying of findbugs.jar and bcel.jar.
3712		daveho	21:37:12, Sonntag, 6. März 2005	Fix to allow compilation now that DetectorFactory no longer
3711		daveho	21:33:02, Sonntag, 6. März 2005	Temporary hack
3710		daveho	21:31:16, Sonntag, 6. März 2005	Detectors are no longer enabled/disabled by setting an ena
3709		daveho	21:14:05, Sonntag, 6. März 2005	Cleaned up imports, added serialVersionUID
3708		daveho	21:13:07, Sonntag, 6. März 2005	Added serialVersionUID
3707		daveho	21:10:09, Sonntag, 6. März 2005	Add serialVersionUID
3706		daveho	19:45:55, Sonntag, 6. März 2005	Modified clone() to call super.clone(), and to use generic ty

Detectors are no longer enabled/disabled by setting an enabled field in the DetectorFactory. Instead, an instance of UserPreferences records which detectors are enabled/disabled.

Action	Path	Copy from path	Revision
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/DetectorFactory.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/DetectorFactoryCollection.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/FindBugs.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/config/ProjectFilterSettings.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/config/UserPreferences.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/gui/AnalysisRun.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/gui/ConfigureDetectorsDialog.java		
Modified	/trunk/findbugs/src/java/edu/umd/cs/findbugs/gui/FindBugsFrame.java		

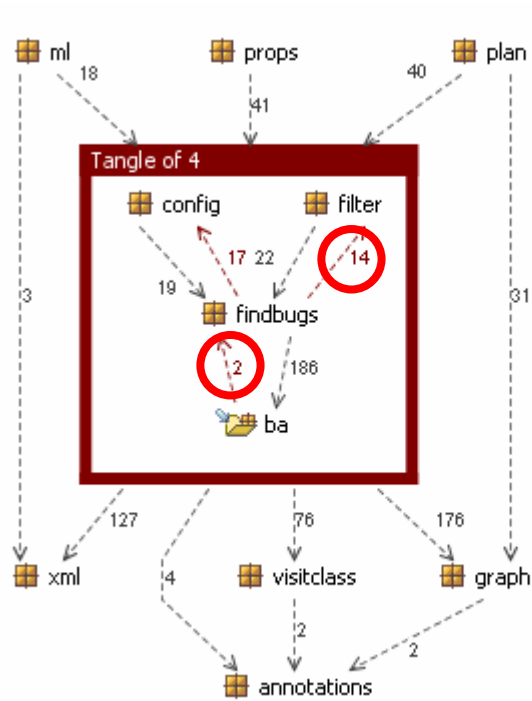
Showing 438 revision(s), from revision 3647 to revision 4084 - 1 revision(s) selected.

Hide unrelated changed paths Stop on copy/rename Include merged revisions

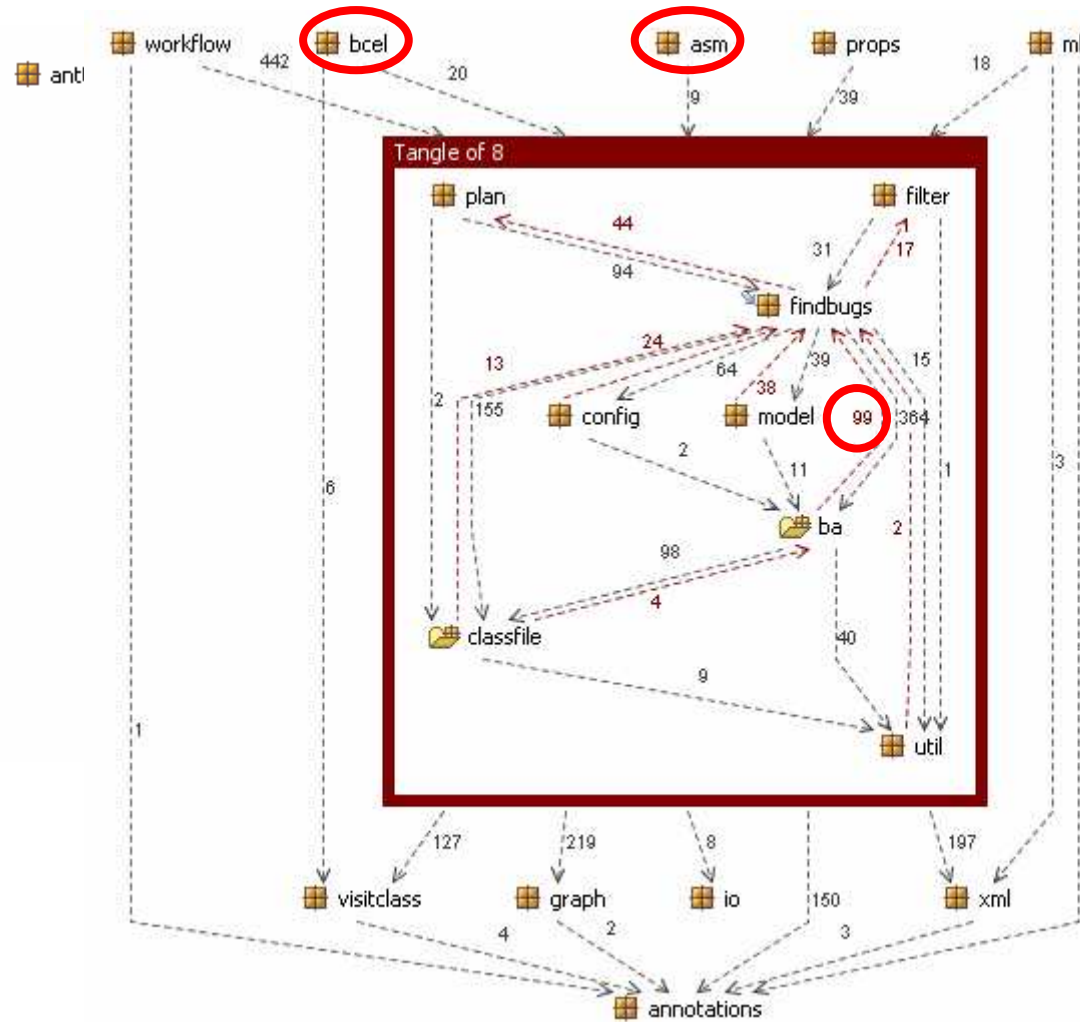
Show All Next 100 Refresh Statistics Help OK

Tangle increase...

SICK

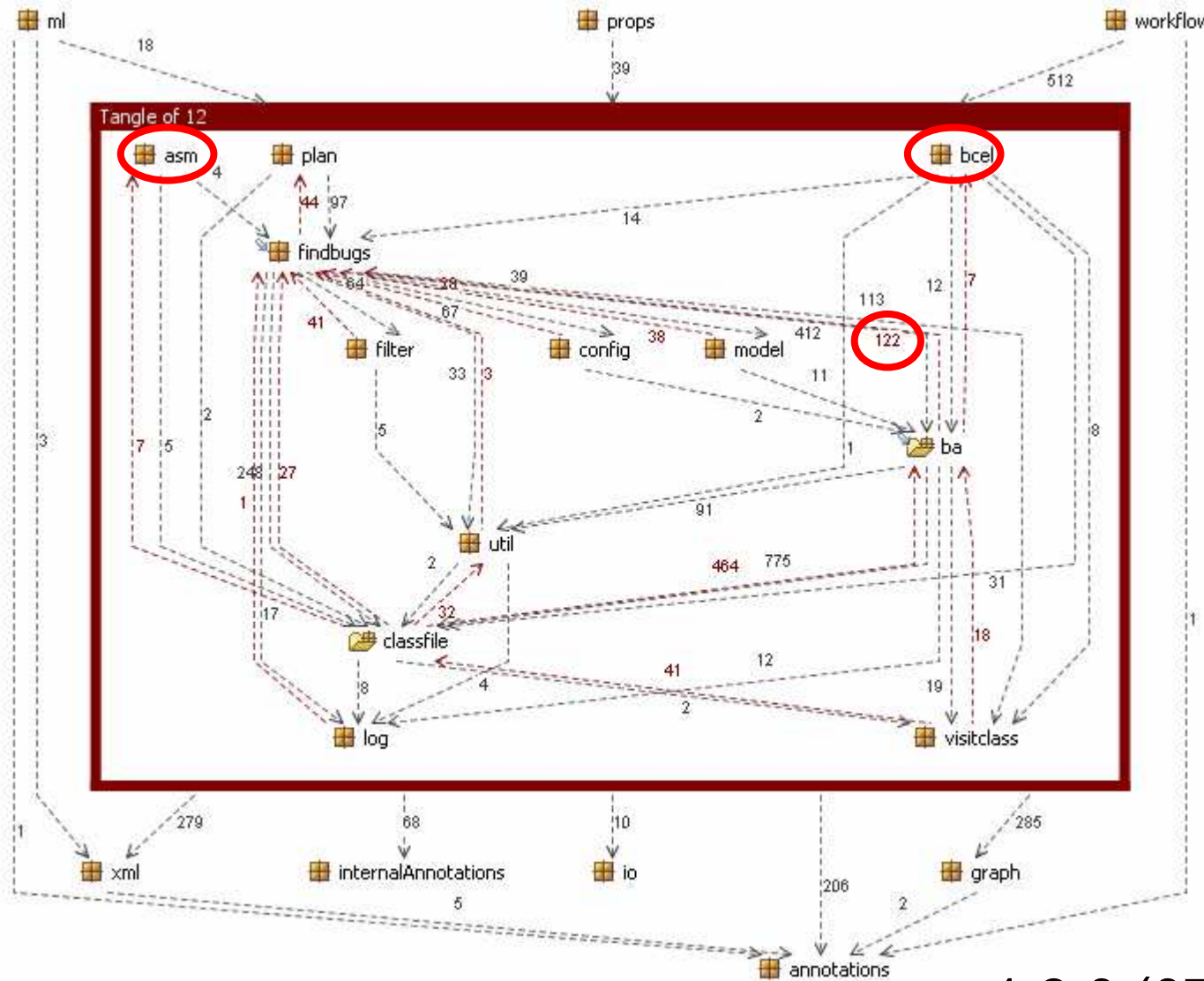


0.8.8 (05/2005)



1.0.0 (06/2006)

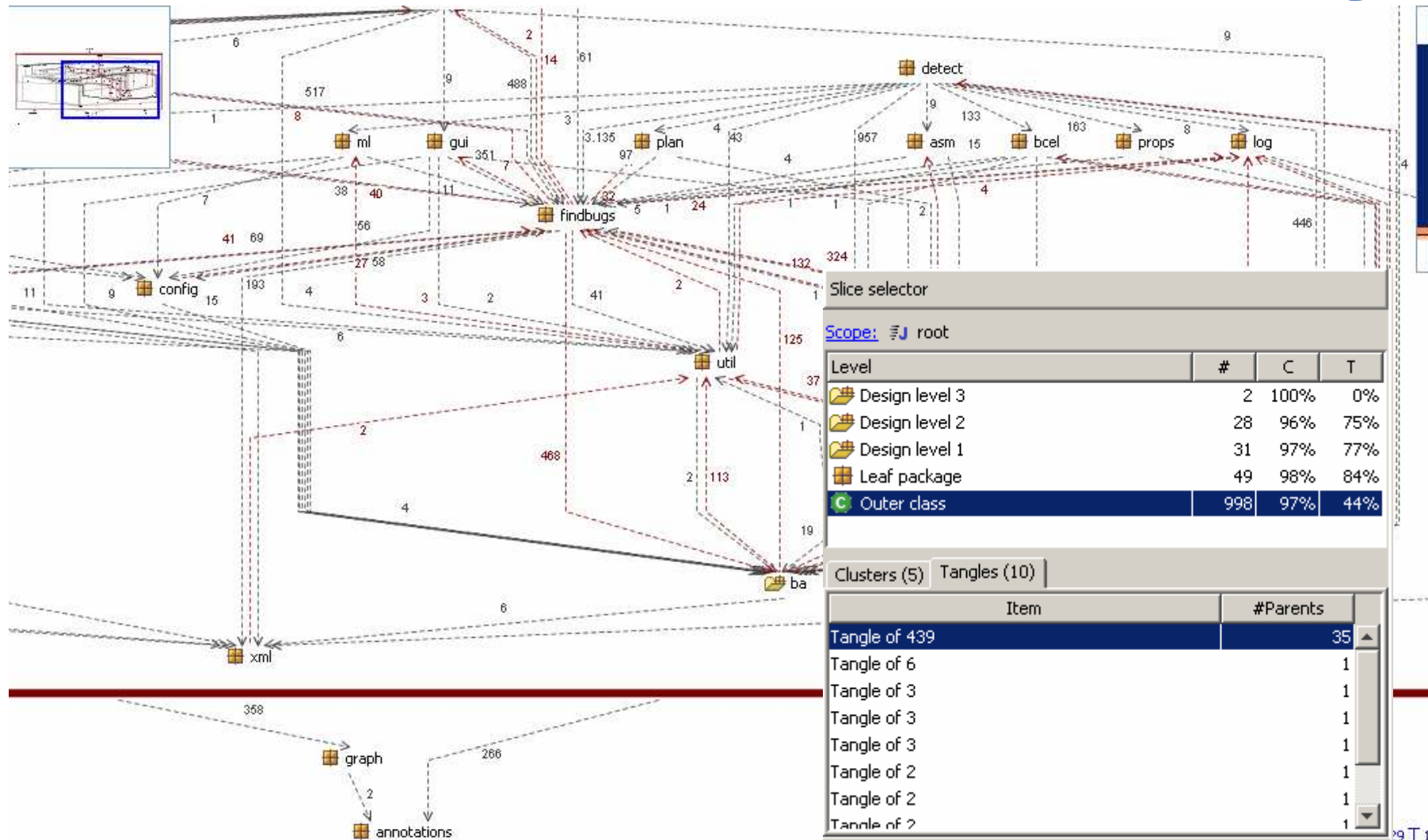
Tangle increase even more...



1.3.0 (07/2007)

ONE BIG Tangle...

SICK



1.3.8 (03/2009)

Findbugs: and the next level of checking ;-)

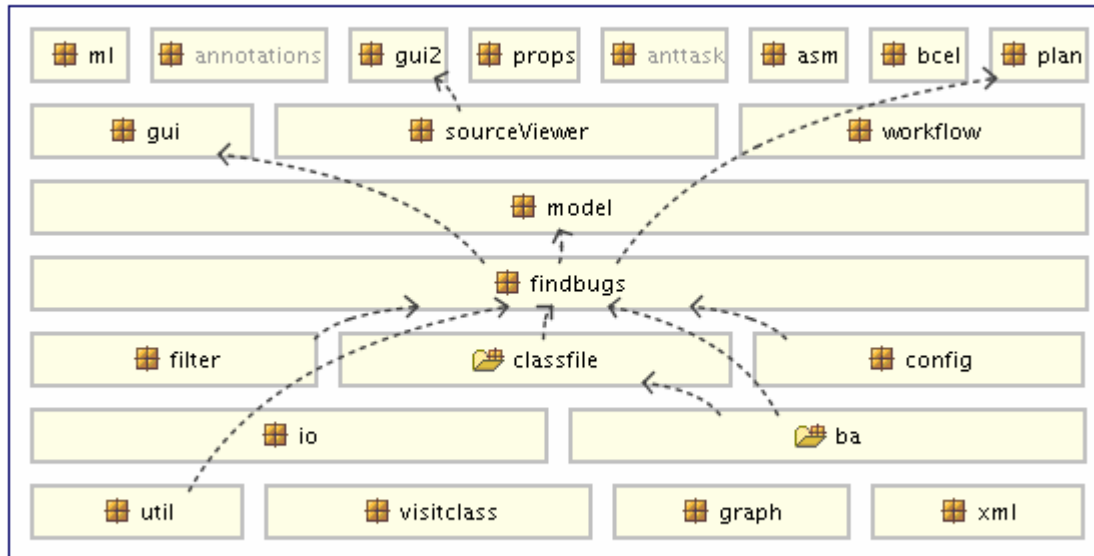
SICK



Findbugs 1.2.1: Architectural Analysis



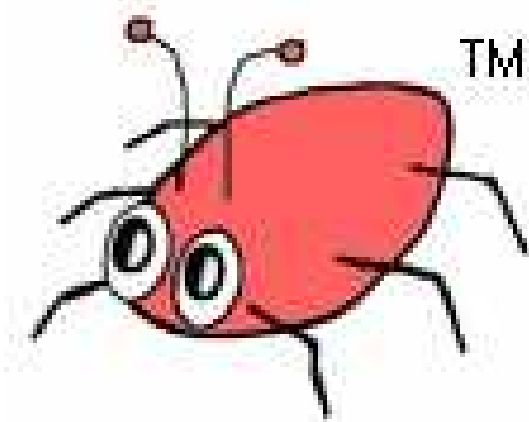
Diagram 1



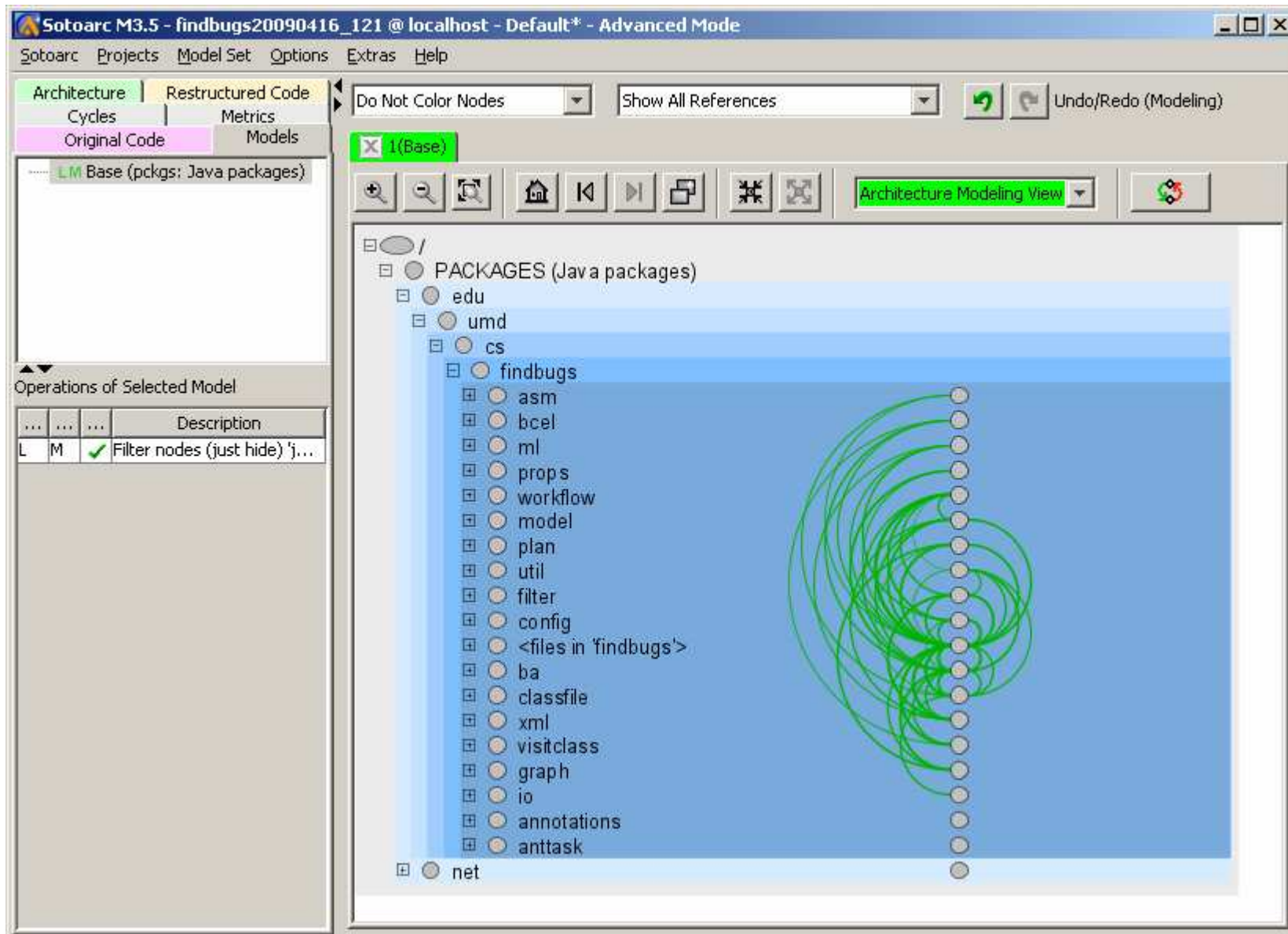
Description: *Subsystem breakout for 'edu.umd.cs.findbugs'*

Violations

Source	Target	#Violations
sourceViewer	gui2	7
findbugs	plan	37
findbugs	gui	7
findbugs	model	30
filter	findbugs	36
classfile	findbugs	12
config	findbugs	19
ba	findbugs	99
ba	classfile	78
util	findbugs	2
Total		327



Modeling Subsystems:



Modeling Subsystems:

The screenshot shows the Sotoarc M3.5 software interface. The main window displays a modeling view of a subsystem structure. The left sidebar shows a tree view of the subsystems, including 'fb.asm', 'fb.bcel', 'fb.ml', 'fb.props', 'fb.workflow', 'fb.model', 'fb.plan', 'fb.util', 'fb.filter', and 'fb.config'. The main area shows a complex network of green lines representing relationships between these subsystems. A tooltip is visible over the 'fb.config' entry, providing details about the creation of unrestricted subsystems.

Operations of Selected Model

...	Description
L	M	✓	Filter nodes (just hide) 'j...
L	M	✓	Create unrestricted subsys...
L	M	✓	Create 'fb.findbugs' (unr...

Create unrestricted subsystems prefixed with 'fb.' beneath '/'
 - content pattern '+ /PACKAGES/edu/umd/cs/findbugs/(*)'
 - base set '***'

 - no privacy pattern
 Resulting sibling order
 - '/fb.asm'
 - '/fb.bcel'
 - '/fb.ml'
 - '/fb.props'
 - '/fb.workflow'
 - '/fb.model'
 - '/fb.plan'
 - '/fb.util'
 - '/fb.filter'
 - '/fb.config'
 - '/fb.ba'
 - '/fb.classfile'
 - '/fb.xml'
 - '/fb.visitclass'
 - '/fb.graph'

Modeling Subsystems:

Sotoarc M3.5 - findbugs20090416_121 @ localhost - findbugs_121 - Advanced Mode

Sotoarc Projects Model Set Options Extras Help

Architecture | Restructured Code
Cycles | Metrics
Original Code | Models

Do Not Color Nodes Show All References Undo/Redo (Modeling)

1(Base)

Architecture Modeling View

- top
 - fb.asm
 - fb.bcel
 - fb.ml
 - fb.props
 - fb.plan
 - fb.annotations
 - fb.anttask
- gui
 - fb.workflow
- model
 - fb.model
- findbugs
 - fb.findbugs
- filter-classfile-config
 - fb.filter
 - fb.config
 - fb.classfile
- io-ba
 - fb.ba
 - fb.io
- util
 - fb.util
 - fb.xml
 - fb.visitclass
 - fb.graph

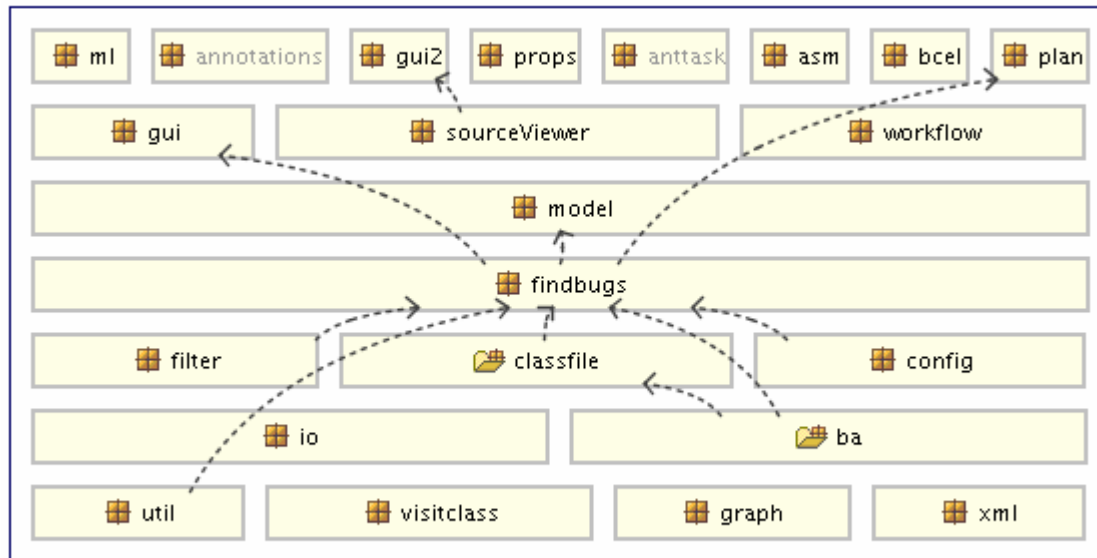
PACKAGES (Java packages)

...	Description
L	M	✓	Filter nodes (just hide) 'j...
L	M	✓	Create unrestricted subs...
L	M	✓	Create 'fb.findbugs' (unr...
L	M	✓	Create 'util' (layer) bene...
L	M	✓	Create 'io-ba' (layer) ben...
L	M	✓	Create 'filter-classfile-co...
L	M	✓	Create 'findbugs' (layer) ...
L	M	✓	Create 'model' (layer) be...
L	M	✓	Create 'gui' (layer) bene...
L	M	✓	Create 'top' (layer) bene...
L	M	✓	Fix child order of '/'

Fixing Architectural Violations

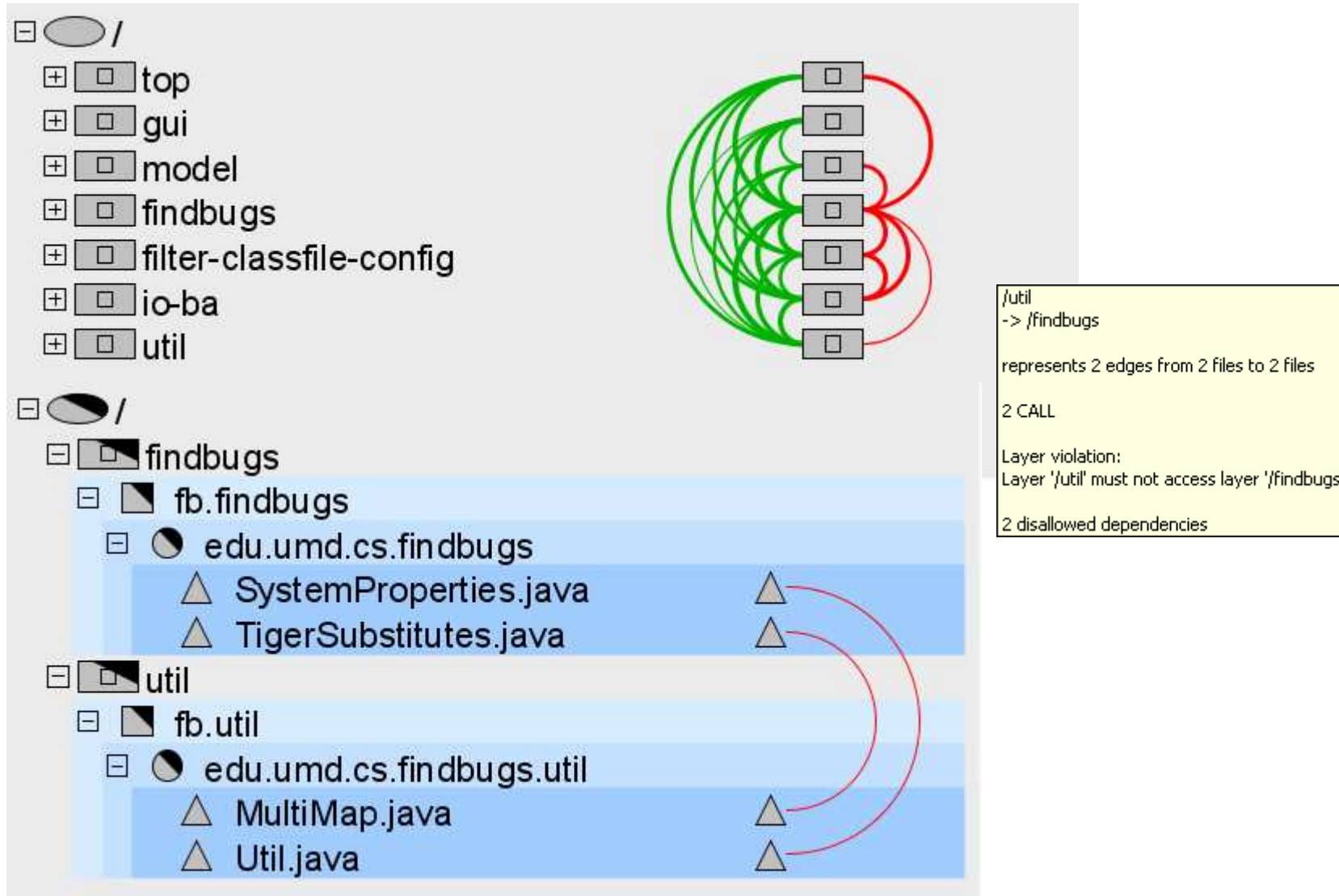


Diagram 1



Description: *Subsystem breakout for 'edu.urnd.cs.findbugs'*

Fixing Architectural Violations



Fixing Architectural Violations



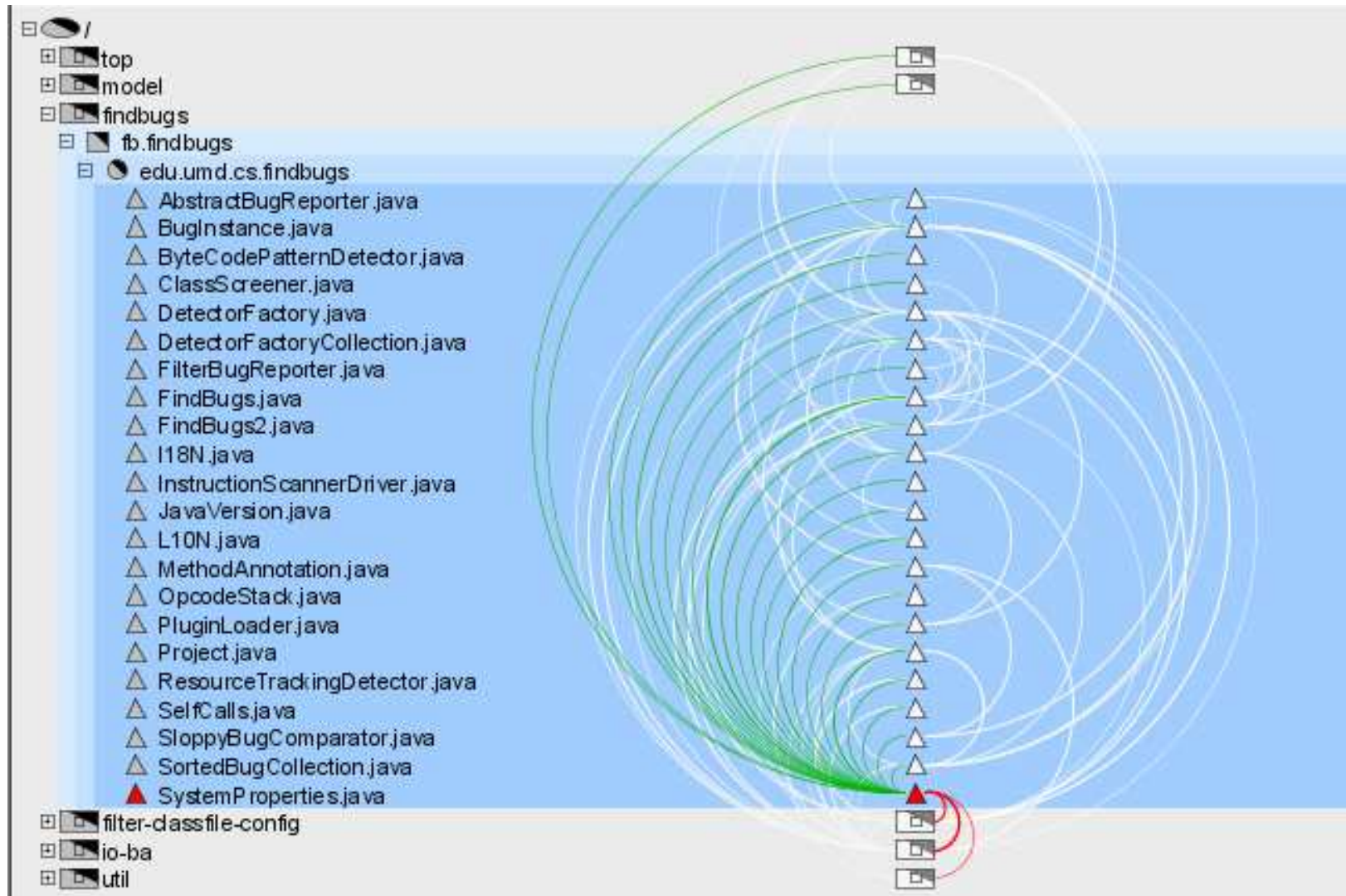
E:\findbugs\src\findbugs-1.2.1\src\java\edu\umd\cs\findbugs\util\Util.java

Filter Remove Filter

ID	ID	fromName	fromKind	ID	fromFile	ID	toName	toKind	ID	toFile	referenceType
PO	SY	{init_Util_attributes}	METHOD	SF	Util.java	SY	getBoolean	METHOD	SF	SystemProperties.java	CALL

```
39 import edu.umd.cs.findbugs.annotations.CheckForNull;
40
41 /**
42  * @author pugh
43  */
44 public class Util {
45 -> public static final boolean LOGGING = SystemProperties.getBoolean("findbugs.shutdownLogging");
46
47     public static void runLogAtShutdown(Runnable r) {
48         if (LOGGING) Runtime.getRuntime().addShutdownHook(new Thread(r));
49     }
50
51
52     public static int nullSafeHashCode(@CheckForNull Object o) {
53         if (o == null) return 0;
54         return o.hashCode();
55     }
56     public static <T> boolean nullSafeEquals(@CheckForNull T o1, @CheckForNull T o2) {
57         if (o1 == o2) return true;
```


Fixing Architectural Violations



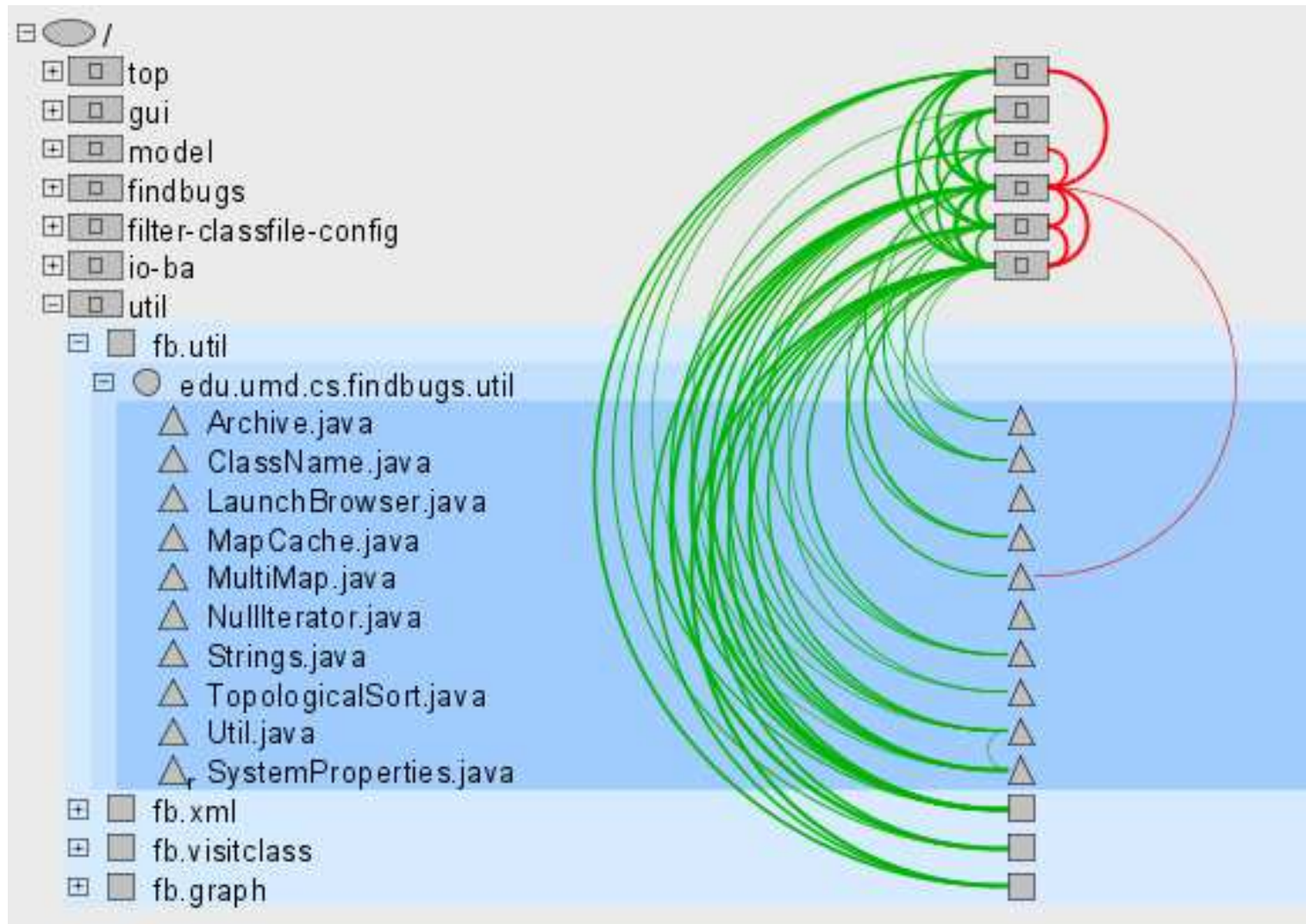
Fixing Architectural Violations

The screenshot shows the Sotoarc M3.5 software interface. The main window displays an architectural model with a tree view on the left and a diagram area on the right. A 'Please Confirm' dialog box is overlaid on the diagram, asking for confirmation to virtually restructure the package tree. The dialog text reads: 'You are attempting to virtually restructure your package tree. This restructuring will be executed before the already defined modeling operations and can, therefore, change their results. Proceed?' with 'Yes' and 'No' buttons.

Below the dialog, the tree view shows the following structure:

- top
 - model
 - InstructionScannerDriver.java
 - JavaVersion.java
 - L10N.java
 - MethodAnnotation.java
 - OpcodesStack.java
 - PluginLoader.java
 - Project.java
 - ResourceTrackingDetector.java
 - SelfCalls.java
 - SloppyBugComparator.java
 - SortedBugCollection.java
 - filter-classfile-config
 - io-ba
 - util
 - fb.util
 - SystemProperties.java
 - Util.java

Fixing Architectural Violations



- Boost

1. do AA-Tools work with heavy templated code ?

2. find out if Boost has some architectural violations: e.g. dependencies

1. from **utility** to other libraries
2. from **somelib** to "heavy" libraries like **mpl**
3. which access the internal implementation details, e.g. (aux_, detail, impl)
4. which violate the existing documented architecture,



Boost: Checking dependencies ?

- Deliver “units”, “iterator_facade”, etc.
- bcp.exe
 - `bcp --list boost\iterator\iterator_facade.hpp > deps.txt`
 - about 300 header files
 - about 20 times mpl usage
 - remove some capability,
 - traits pod, const, etc
 - no need to include std mpl...
 - no support for `iterator_writability_disabled<>`

Organization - Mozilla Firefox

File Edit View History Bookmarks ScrapBook Tools Help

file:///B:/boost_1_37_0/libs/spirit/classic/doc/organization.html

Organization

↑ ← →

The framework is highly modular and is organized in layers:

```
graph TD; L1[iterator actor]; L2[debug]; L3[attribute dynamic error_handling symbols tree utility]; L4[meta]; L5[scanner primitives composite non_terminal]; L1 --- L2 --- L3 --- L4 --- L5;
```

Build Setup

- Usually capture process
- header only (90%)
- Idea
 - Compile test cases (libs)

```
$ cd boost_1_37_0
$ find boost/mpl -name "*.hpp" | grep -v aux_ > includes.txt
$ for i in `cat includes.txt` ; do printf "#include <${i}>\n";
done >mplmain.cpp
```

```
$ cafeCC -BB:\boost_1_37_0 -I. -shared -w -no_strip -o
mpl.dll.iml mplmain.cpp
```

```
$ iml2rfg -lifted_module_view mpl.dll.iml mpl.dll.rfg
```

SA: Modeling Boost

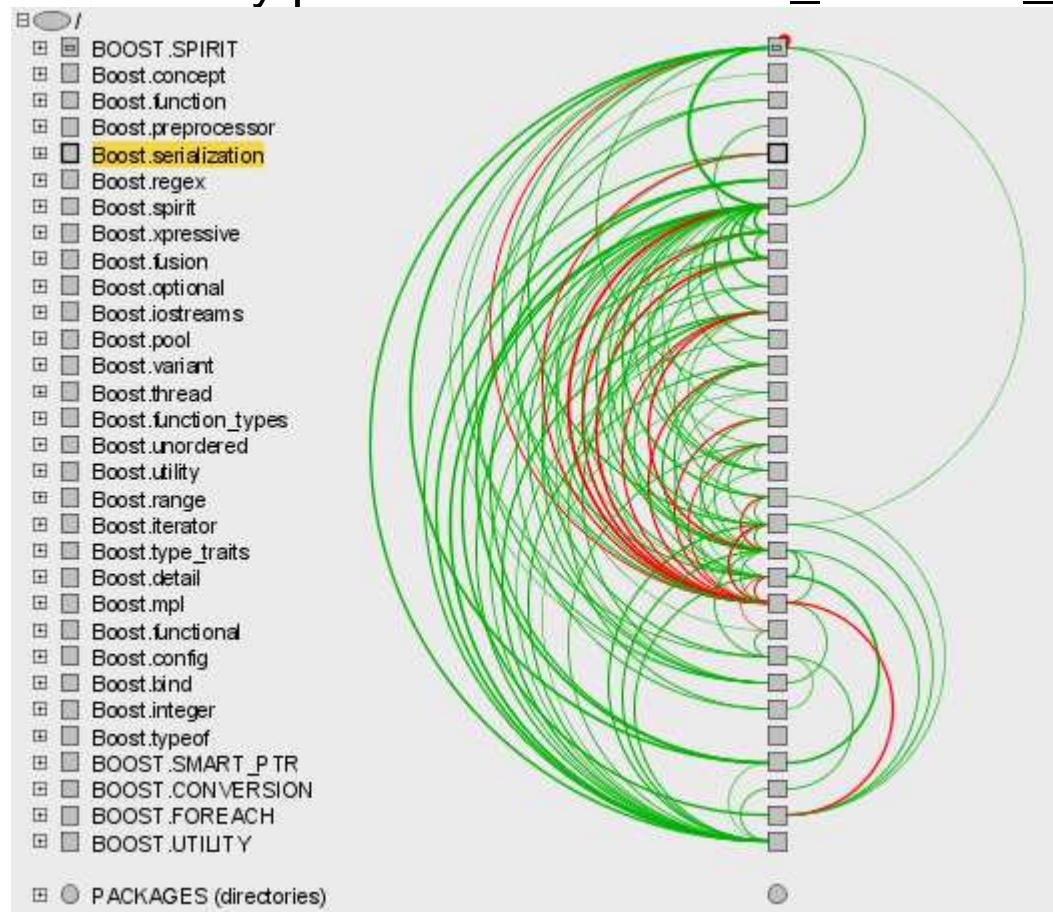
L	M	✓	Create 'BOOST.SPIRIT' (unrestricted subsystem) benea...
L	M	✓	Move 'IteratorActor', 'Debug', 'AttributeDynamicErrorSy...
L	M	✓	Fix child order of 'BOOST.SPIRIT'
L	M	✓	Create 'BOOST.UTILITY' (unrestricted subsystem) bene...
L	M	✓	Move 'utility' to 'BOOST.UTILITY'
L	M	✓	Create 'BOOST.SMART_PTR' (unrestricted subsystem) ...
L	M	✓	Move 'smart_ptr' to 'BOOST.SMART_PTR'
L	M	✓	Create 'BOOST.CONVERSION' (unrestricted subsystem)...
L	M	✓	Move 'conversion' to 'BOOST.CONVERSION'
L	M	✓	Filter nodes (just hide) 'optional.hpp'
L	M	✓	Filter nodes (just hide) 'config.hpp'
L	M	✓	Filter nodes (just hide) 'unordered_map.hpp'
L	M	✓	Create 'BOOST.FOREACH' (unrestricted subsystem) be...

- BOOST.SMART_PTR
- BOOST.CONVERSION
- BOOST.FOREACH
- BOOST.UTILITY

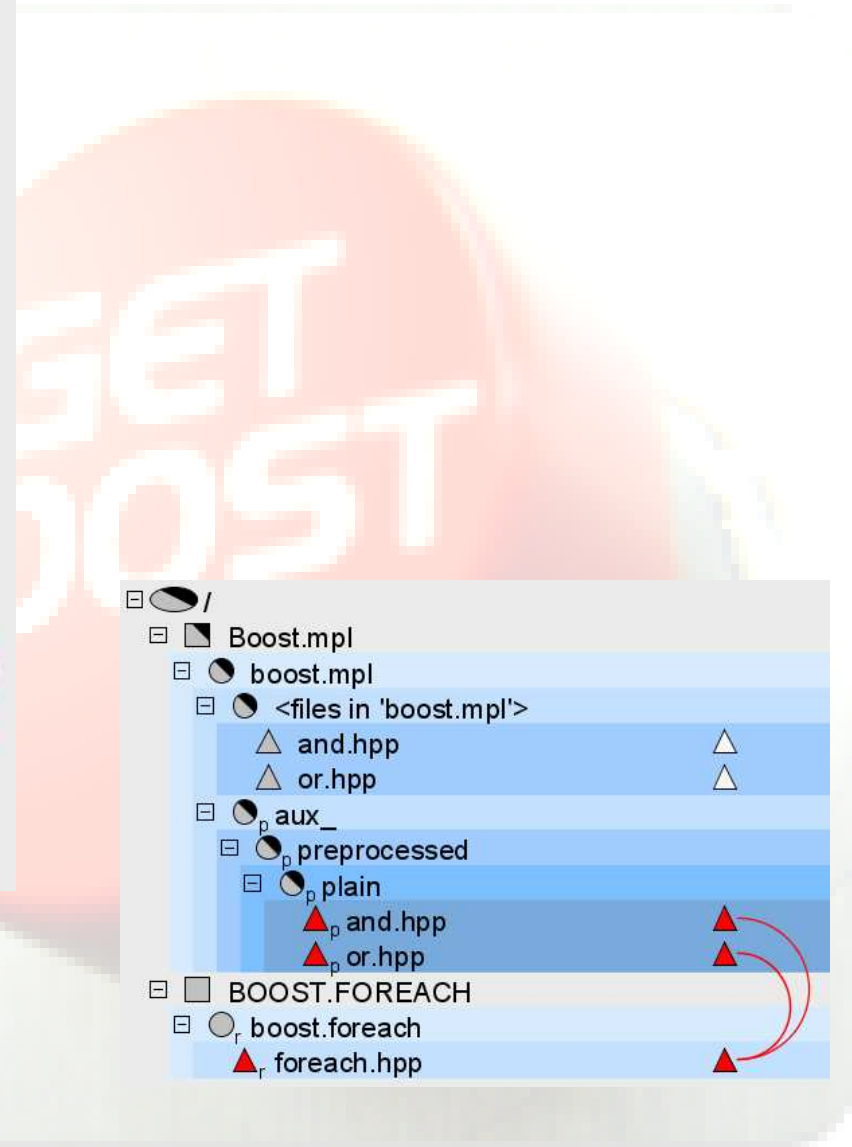
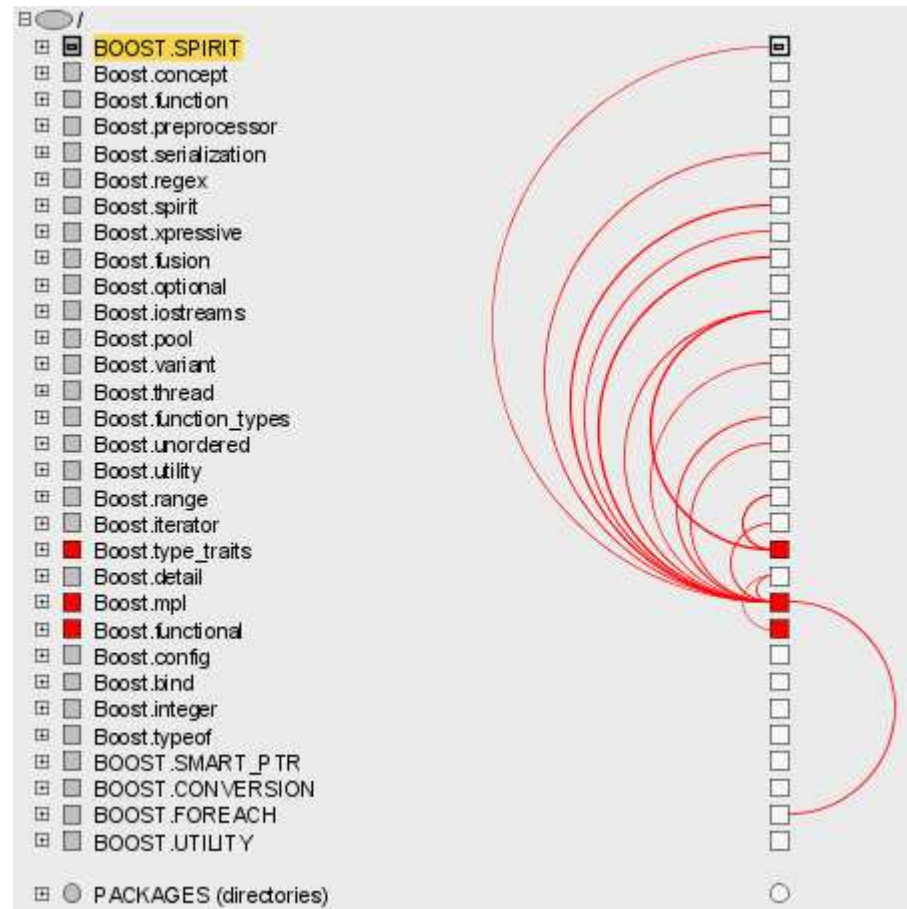
...	Description
L	R	✓	Create 'utility' (container) beneath PACKAGES
L	R	✓	Move 'assert.hpp', 'call_traits.hpp', 'checked_delete.hp...
L	R	✓	Move 'integer_traits.hpp' to 'integer'
L	R	✓	Move 'range.hpp' to 'range'
L	R	✓	Move 'regex.hpp', 'regex_fwd.hpp' to 'regex'
L	R	✓	Move 'type_traits.hpp' to 'type_traits'
L	R	✓	Move 'iterator_adaptors.hpp' to 'iterator'
L	R	✓	Move 'preprocessor.hpp' to 'preprocessor'
L	R	✓	Create 'smart_ptr' (container) beneath PACKAGES
L	R	✓	Move 'intrusive_ptr.hpp', 'scoped_array.hpp', 'scoped_...
L	R	✓	Move 'mem_fn.hpp' to 'bind'
L	R	✓	Move 'function.hpp' to 'function'
L	R	✓	Create 'conversion' (container) beneath boost
L	R	✓	Move 'implicit_cast.hpp', 'lexical_cast.hpp' to 'conversion'
L	R	✓	Move 'blank.hpp', 'blank_fwd.hpp' to 'mpl'
L	R	✓	Move 'none.hpp', 'none_t.hpp' to 'optional'
L	R	✓	Move 'memory_order.hpp' to 'smart_ptr'
L	R	✓	Move 'aligned_storage.hpp' to 'detail'
L	R	✓	Move 'array.hpp' to 'utility'
L	R	✓	Create 'foreach' (container) beneath boost
L	R	✓	Move 'foreach.hpp' to 'foreach'
L	R	✓	Move 'ref.hpp' to 'bind'

SA: Modeling Boost

- Create unrestricted subsystems with name X for packages under (Packages)/boost/(X)
- Privacy pattern used + `**/aux_` + `**/aux_/**` + `**/detail` + `**/detail/**`

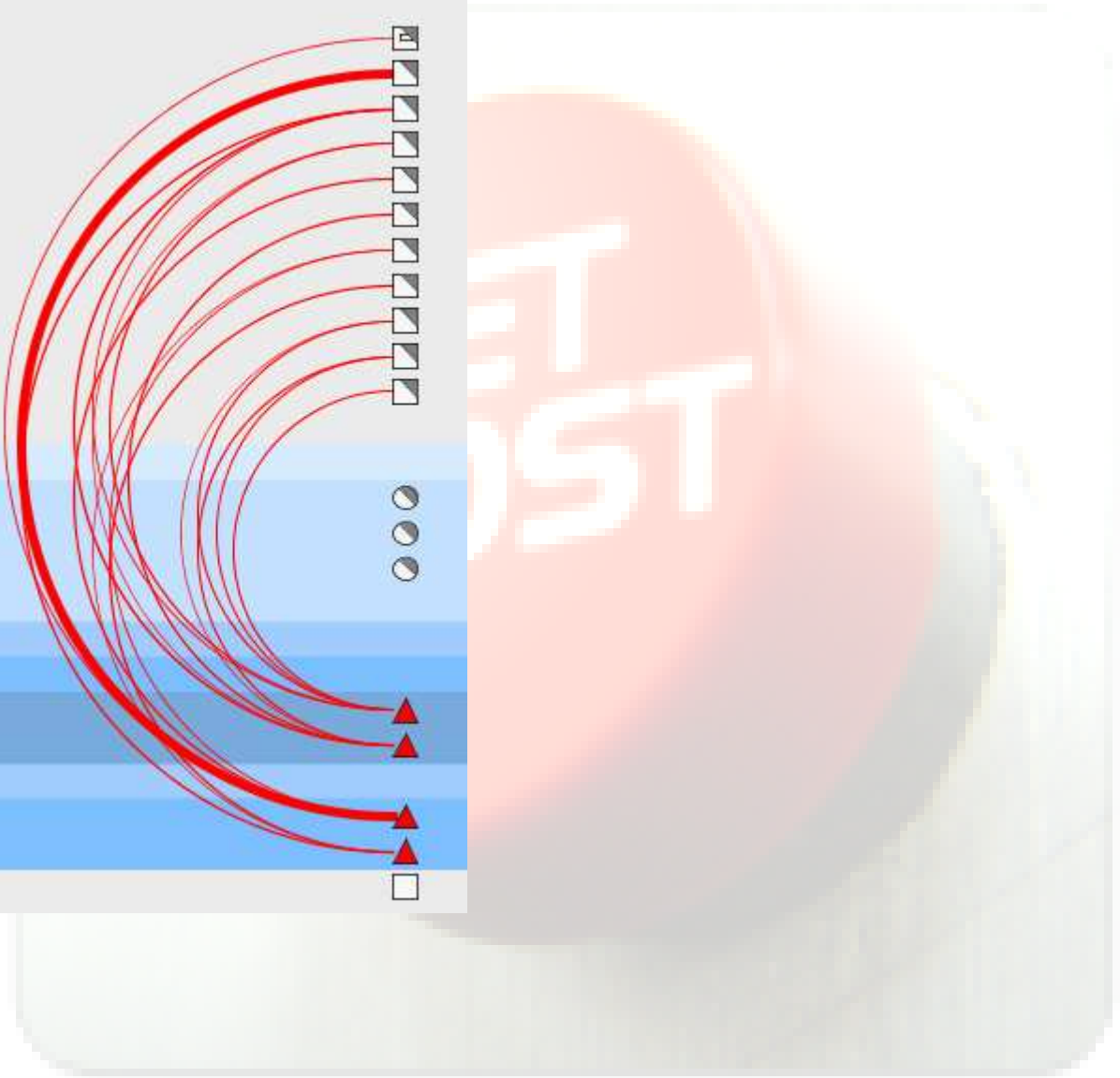
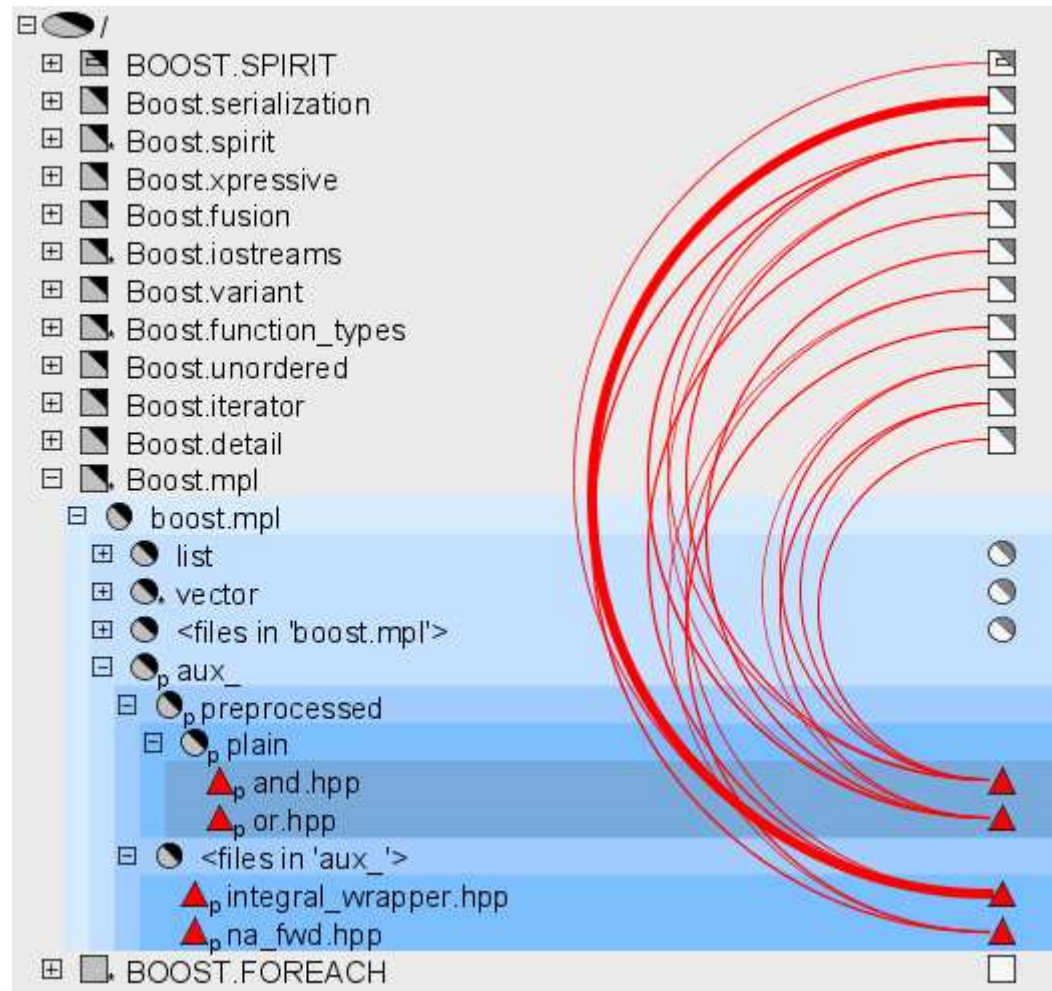


SA: Modeling Boost

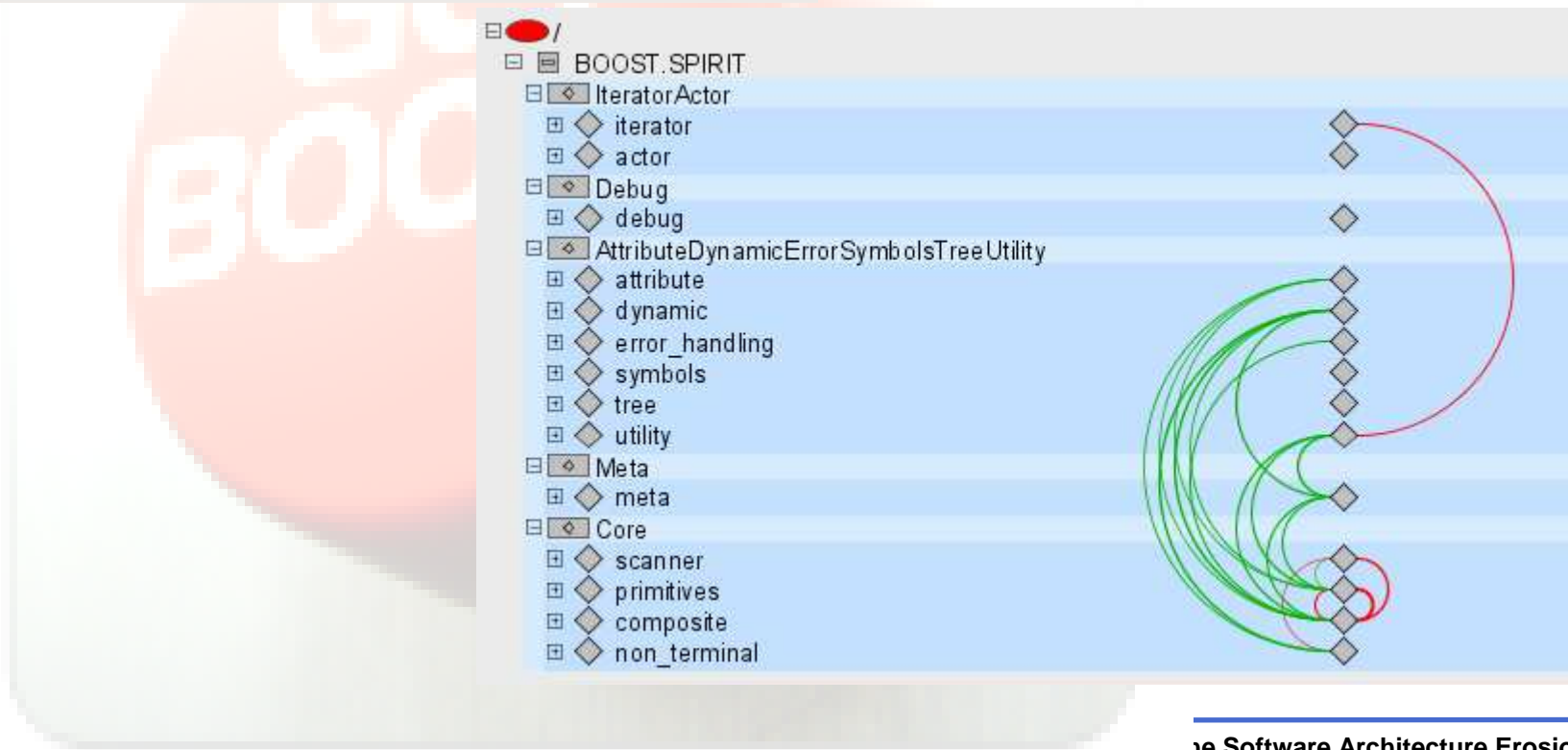
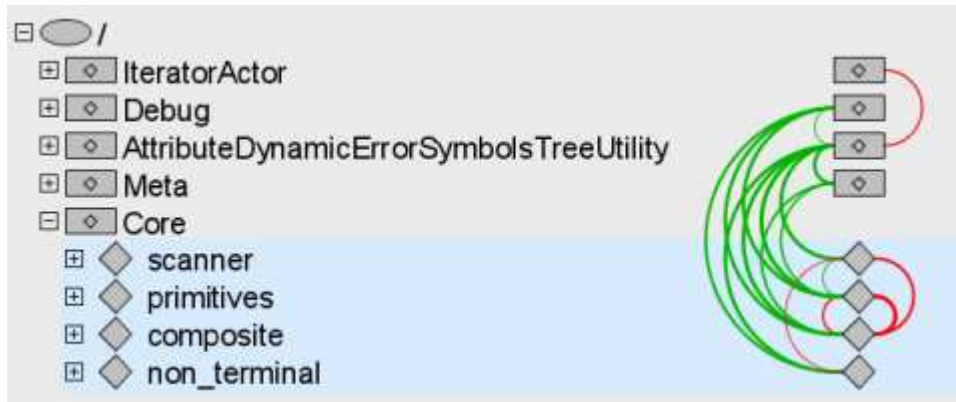


SA: Modeling Boost

SICK



SA: Modeling Spirit

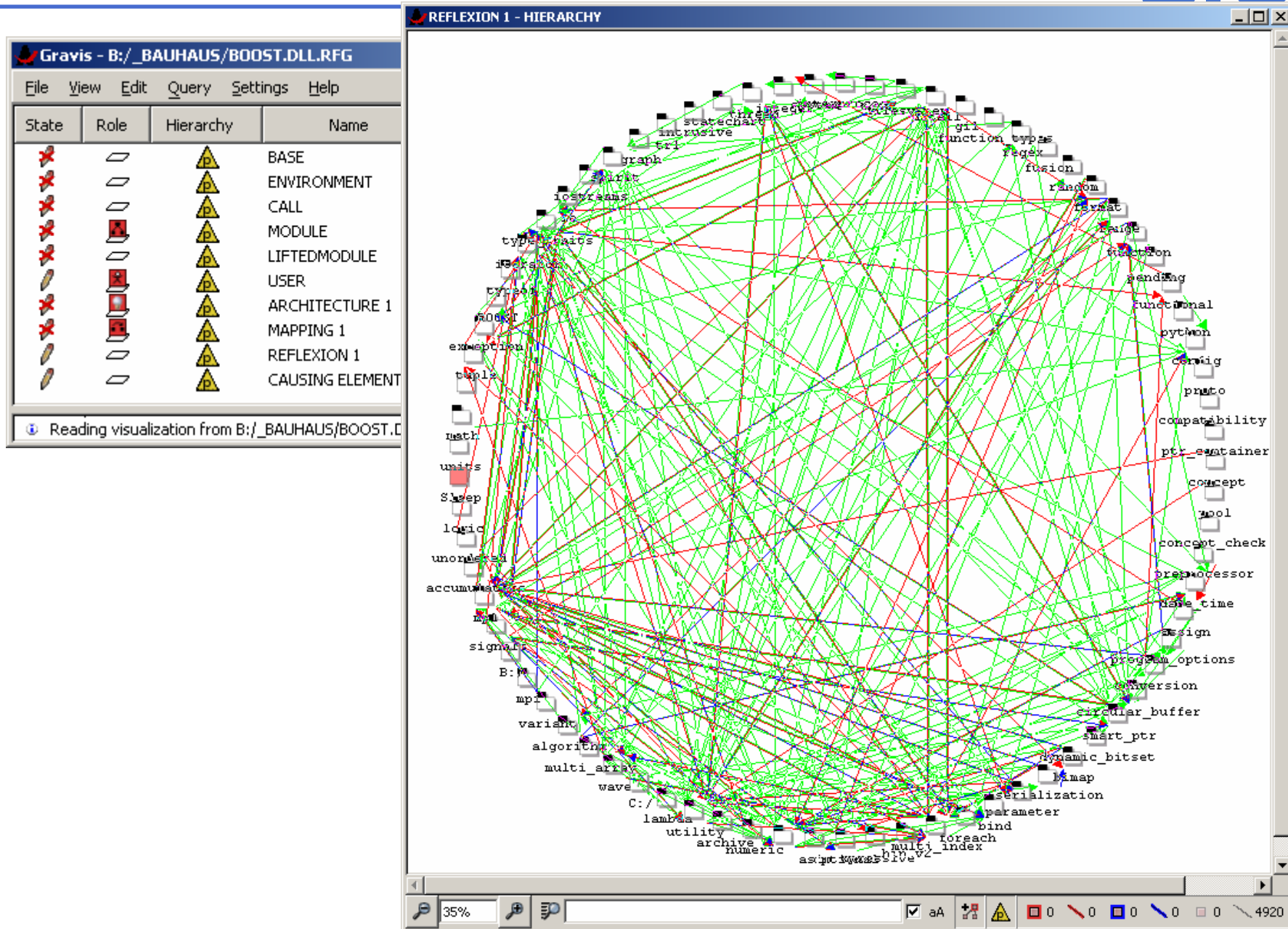


Bauhaus: Modeling



```
B:\_bauhaus\architecture.py - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
architecture.py
194 publicnodes = []
195
196 # 1st step: create architecture elements for each directory
197 for package in (d for d in yield_directory (base)
198                 if os.path.isdir (os.path.join (base, d))):
199     parent = create_node (graph, architecture_view, 'Cluster', package, 'C BOOST/%s' % package.upper ())
200     private = create_node (graph, architecture_view, 'Cluster', package.upper ()+':private', 'C BOOST/%s/PRIVATE' % (package.upper ()))
201     public = create_node (graph, architecture_view, 'Cluster', package.upper ()+':public', 'C BOOST/%s/PUBLIC' % (package.upper ()))
202
203     create_edge (graph, architecture_view, 'Enclosing', private, parent)
204     create_edge (graph, architecture_view, 'Enclosing', public, parent)
205
206     # create allowed relationships
207     create_edge (graph, architecture_view, 'Source_Dependency', private, public)
208     create_edge (graph, architecture_view, 'Source_Dependency', public, private)
209     publicnodes.append(public)
210
211
212
213 # 2nd step: create architecture elements for header files
214 print "Die nachfolgenden Header werden noch nicht behandelt"
215 for package in (d for d in yield_directory (BASE_DIRECTORY)
216                 if d.endswith ('.hpp')):
217     if package in EXCEPTIONS:
218
219     else:
220
221
222
223
224
225
226
227 # 3rd step: create mapping
228 # boost/x --> X.public
229 # boost/y/**/z --> Y.private for z\in PRIVATE_NAMES
230 # boost/h.hpp --> EXCEPTIONS[h].public
231 module_desc = graph.node_type ('Module')
232 dir_desc = graph.node_type ('Directory')
233 for node in bauhaus.rfg.View (graph, 'MODULE').nodes (lambda n: n.is_of_type (module_desc) or n.is_of_type (dir_desc)):
234     if node.is_of_type (dir_desc):
235
236     else:
237
238
239
240
241
242
243
244 # create allowed relationships for all public nodes... this results also in a lot of "absent" relationships
245 for node1 in publicnodes:
246     for node2 in publicnodes:
247         if (node1 == node2):
248             continue
249         create_edge (graph, architecture_view, 'Source_Dependency', node1, node2)
250
251
252
Python file nb char : 13002 Ln : 1 Col : 1 Sel : 0 Dos\Windows ANSI INS
```

Bauhaus: Modeling



No (real) violations found in MPL stuff...

SICK



Joel Childs

Factorial example:

```
template <unsigned N>
struct Factorial
{
    static const unsigned value = N * Factorial<N-1>::value;
};
```

```
template <>
struct Factorial<0>
{
    static const unsigned value = 1;
};
```

```
void main()
{
    int y = Factorial<2>::value;
}
```


- Explicit specialization :

- Factorial<0> is an explicit specialization of Factorial<N> #2
- Factorial<1> is a specialization of Factorial<N> #1

- Function main uses the specialization Factorial<2> #3
- Factorial<2> is a specialization of Factorial<N> #3

- Factorial<2> uses the specialization Factorial<1> #1
- Factorial<1> uses the specialization Factorial<0> #1

Partial specialization:

```
template <typename T>  
class sortedVector  
{...}
```

```
template <typename T>  
class sortedVector<T *>  
{...}
```

```
sortedVector<int >      i1;  
sortedVector<int *>    i2;  
sortedVector<long>     j1;  
sortedVector<long *>  j2;
```

Template Dependencies:

Structure101 for QAC and QAC++ - B:_s101\test.qac.hsp

File Model Navigate Tag Tools Perspective Help

Composition hierarchy

- FactorialU.hpp
- FactorialN.hpp
- Factorial_all.hpp
- Factorial_func.hpp
- main.cpp
- partial_specializati...
- (template-instances)
- ::Factorial<0>
- ::Factorial<1>
- ::Factorial<2>
- ::Factorial<3>
- ::Factorial<4>
- ::Factorial_All<0>
- ::Factorial_All<1>
- ::Factorial_All<2>
- ::Factorial_All<3>
- ::Factorial_All<4>

Design tangles (1) Notables

Item	Depth
root	0

Graph contents Scope

Items	Dependencies
From	U... To ...

Dependency graph: (template-instances)

Show as

	::Factorial<0>	::Factorial<1>	::Factorial<2>	::Factorial<3>	::Factorial<4>	::Factorial_All<0>	::Factorial_All<1>	::Factorial_All<2>	::Factorial_All<3>	::Factorial_All<4>	::base<{s::derived}>	::sortedVector<nI>	::sortedVector<nI>	::sortedVector<nI>	::sortedVector<nI>	::sortedVector<nI>
::Factorial<0>	■															
::Factorial<1>		■														
::Factorial<2>			■													
::Factorial<3>				■												
::Factorial<4>					■											
::Factorial_All<0>						■										
::Factorial_All<1>							■									
::Factorial_All<2>								■								
::Factorial_All<3>									■							
::Factorial_All<4>										■						
::base<{s::derived}>											■					
::sortedVector<nI>												■				
::sortedVector<nI>													■			
::sortedVector<nI>														■		
::sortedVector<nI>															■	
::sortedVector<nI>																■

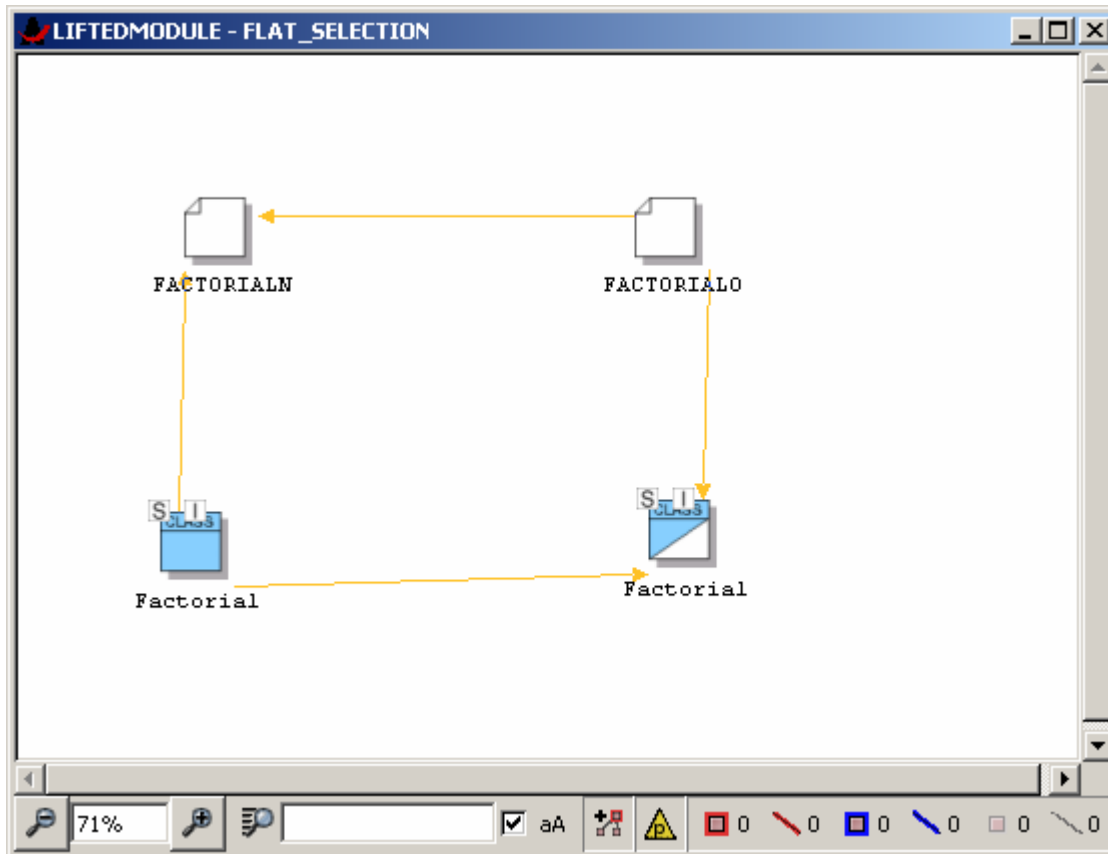
Fit Cell size: 116 Do To

Dependency breakout

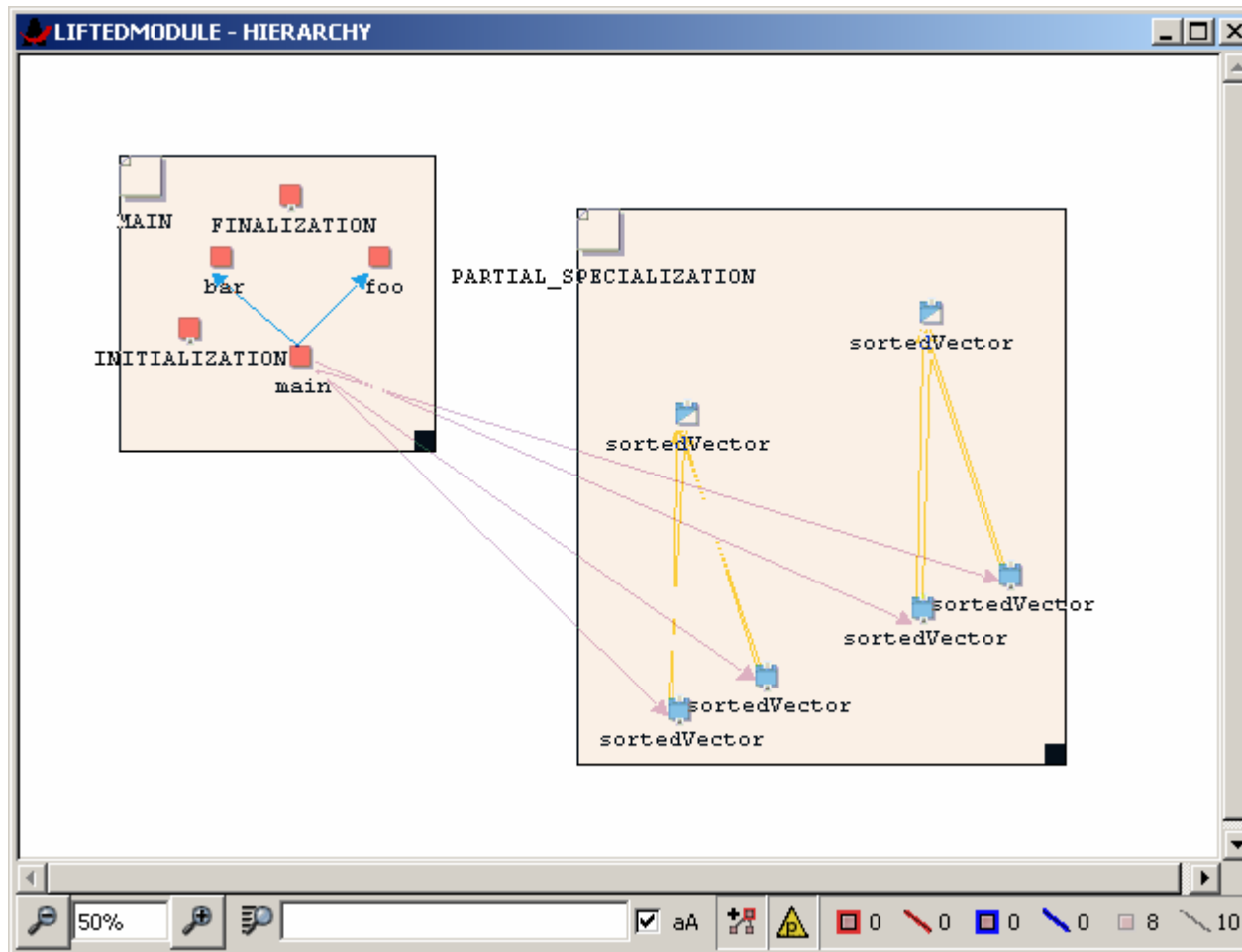
Select a dependency in the graph (or a cell in the matrix) to view its breakout

Transformations: 0

Template Dependencies:



Template Dependencies:



Decuducable dependencies

- Code
 - Usage/Call graph at runtime
- Templates:
 - Usage/Call graph at COMPILE time

- Problems
 - Compiler vs. Analysis tools
 - Optimization vs. verbose information
- C++ Parsers
 - EDG optimizes information away (< EDG 4.0)
 - QA/C++ PRL has some more information

Boost MPL and Architectural Analysis Tools...

SICK



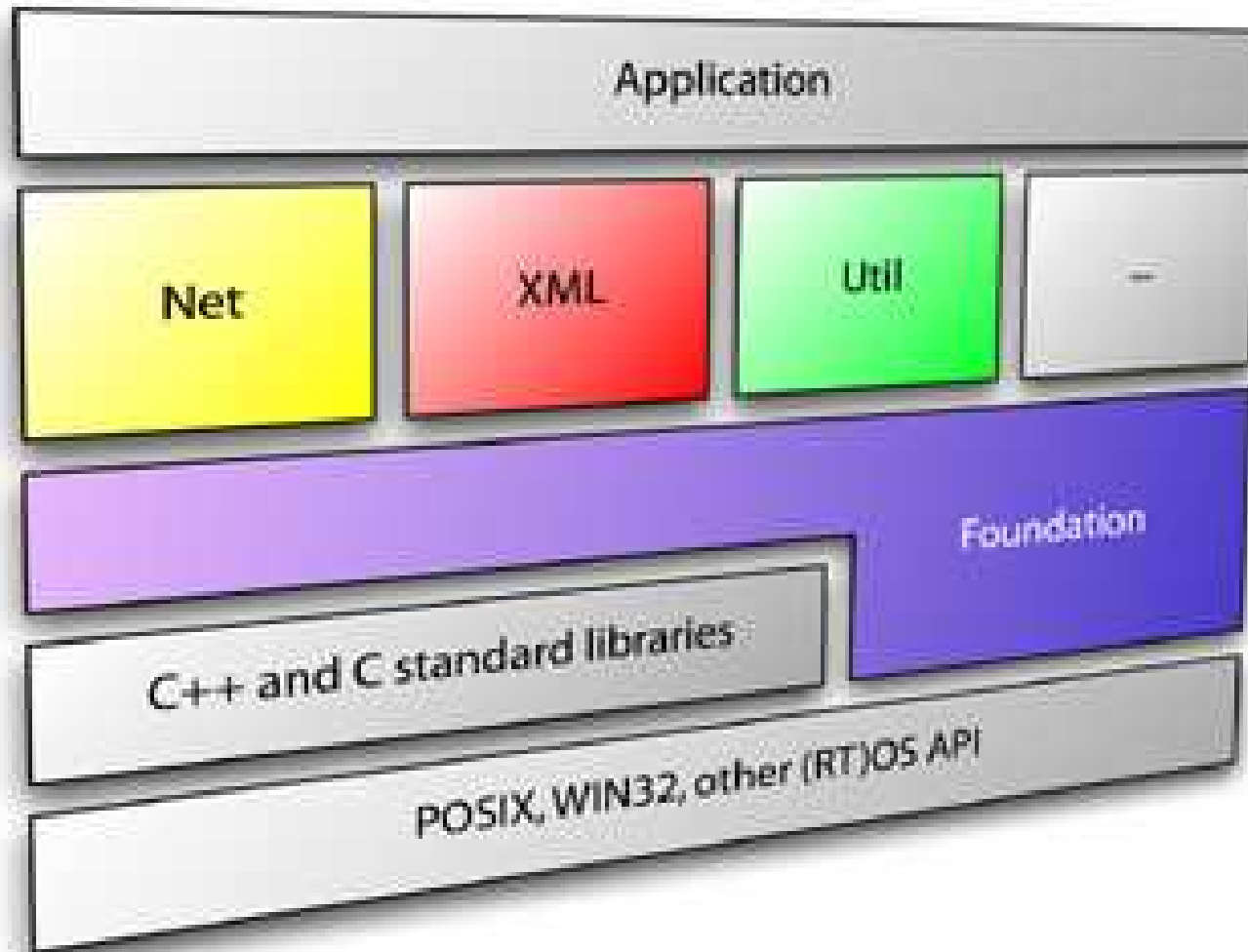
Features

- threads, MT abstractions
- streams and filesystem access
- shared libraries and class loading
- powerful logging and error reporting
- security
- network programming
- XML parsing and generation
- configuration file and options handling
- SQL database access

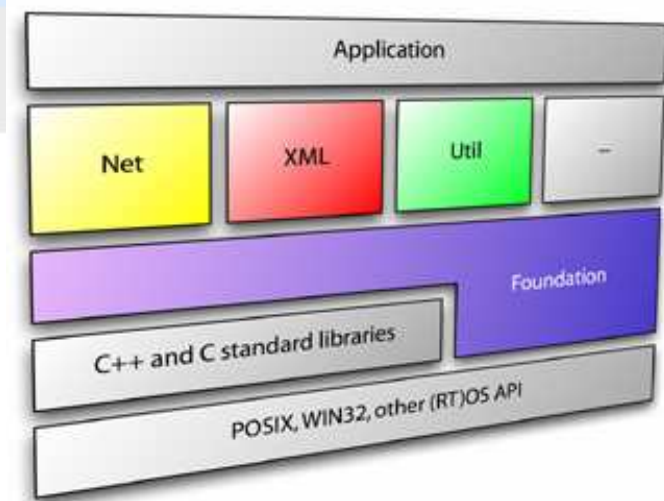
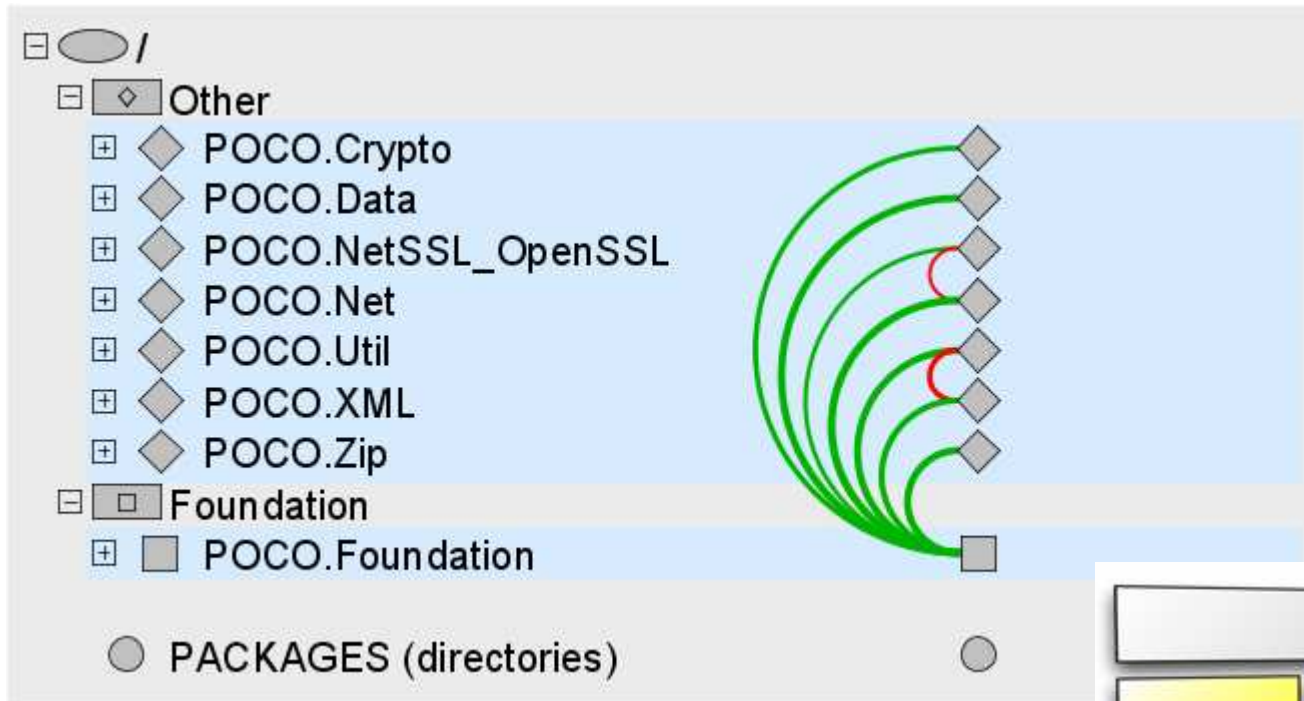


Written by Guenter Obiltschnig , a ACCU member and friend

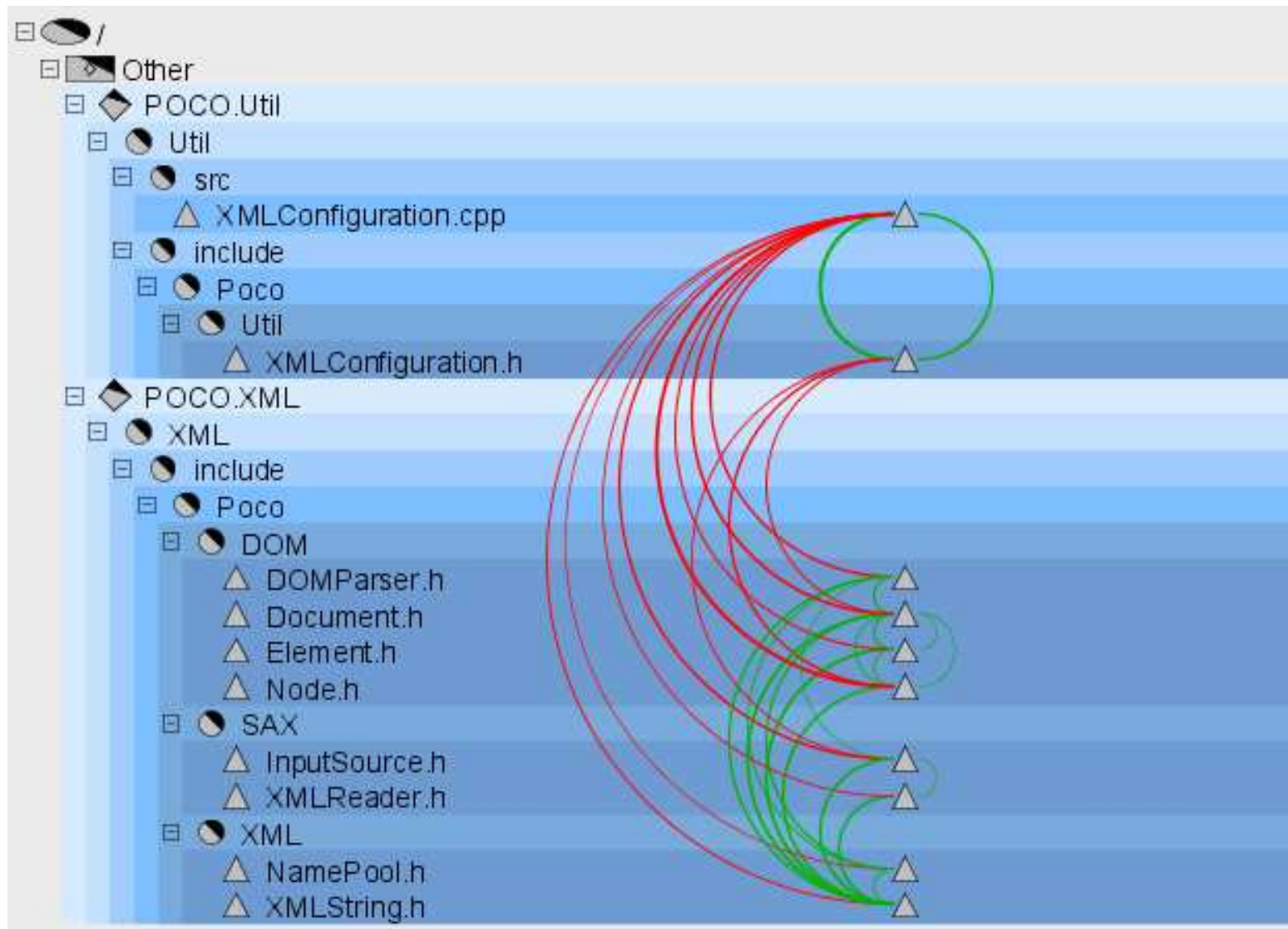
POCO : Should Architecture



POCO 1.3.3: Overall Layering



POCO 1.3.3: Violation Util → XML



MARCH 11, 2009

POCO on Tiny Hardware

Filed under: [C++](#), [Development](#), [Fun](#) by guenter at 20:52

We recently got our hands on a [Digi Connect ME 9210](#), one of the smallest Linux-capable embedded computers in the world. The system, which is just a bit larger than an Ethernet RJ-45 socket, is based on an ARM9 CPU running at 75 MHz. With 4 MB of Flash and 8 MB of DRAM, the system is powerful enough to run POCO-based applications. For example, we ported the [Mindstorms/iPhone](#) controller application from the demo we



showed at Embedded World in Nuremberg to the Connect ME, and it runs great. Well, porting is a bit overstating, as we merely had to build a new Flash image for the Connect ME, and update the application's config file. Well, a 75 MHz ARM9 CPU provides enough power to run an application with a built-in web server powering an Ajax-enabled website. Also, the performance improvements for the 1.3.4 release help a lot to make the

application work great. Additionally, the 1.3.4 release will introduce some minor changes to help reduce the executable size of statically linked applications. For example, it is possible to build the Util library without XMLConfiguration support, which prevents the XML library from being linked in, cutting about 500K from the executable size.

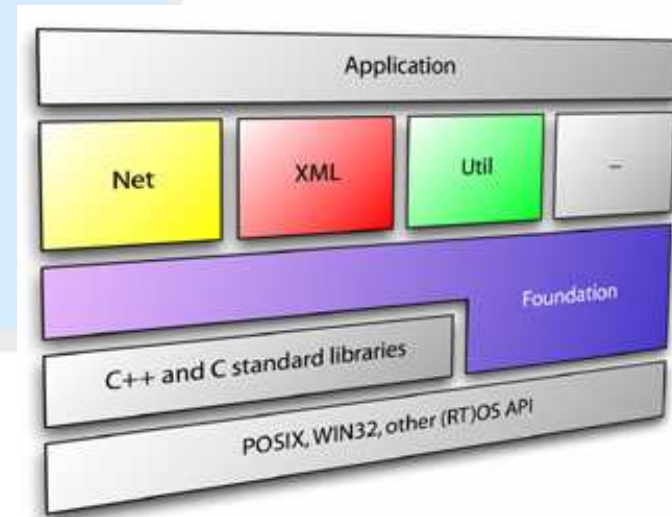
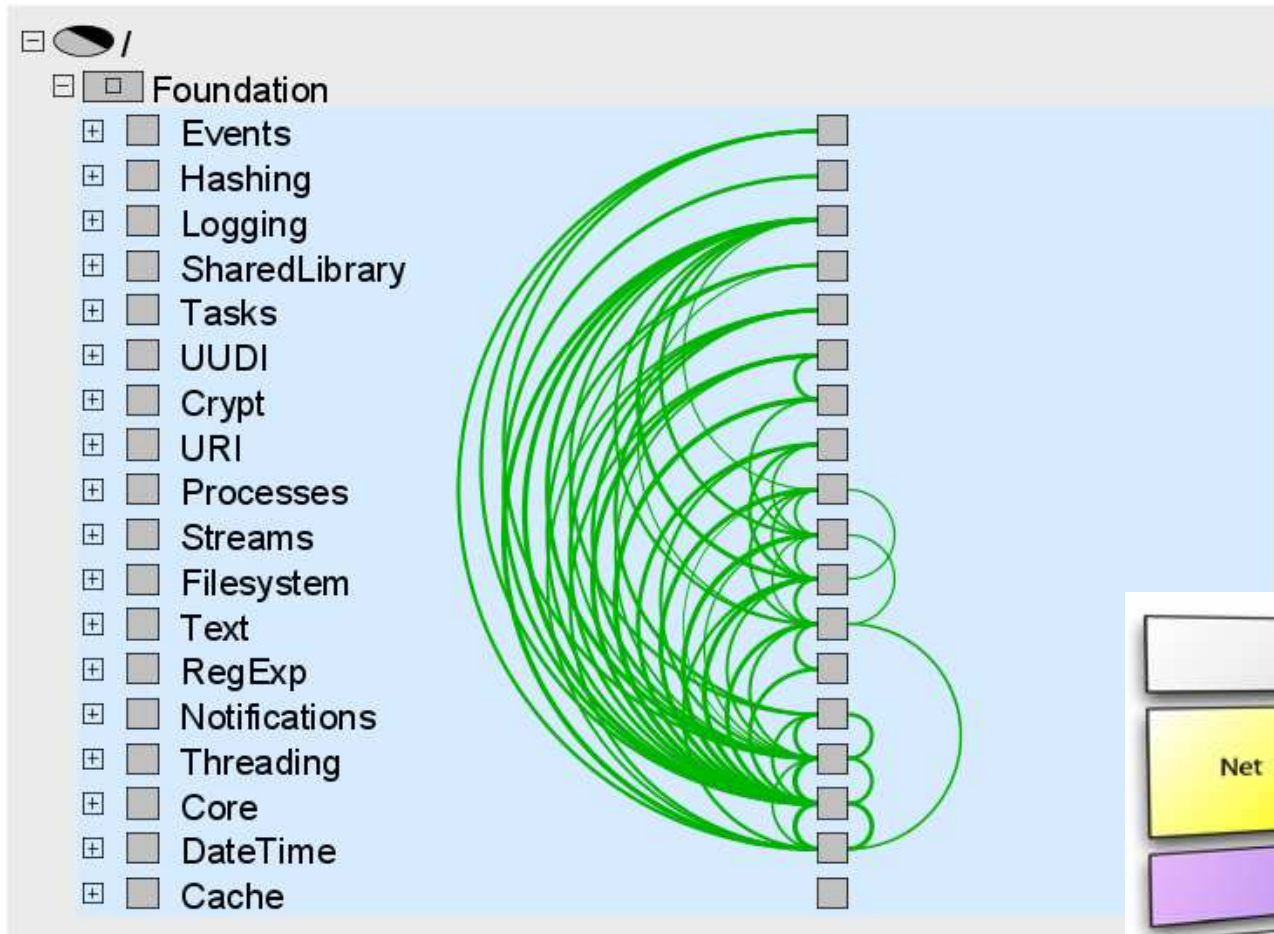
Working with this little device has been a lot of fun, and we are looking forward doing some cool projects with it.

Poco: Architectural Analysis

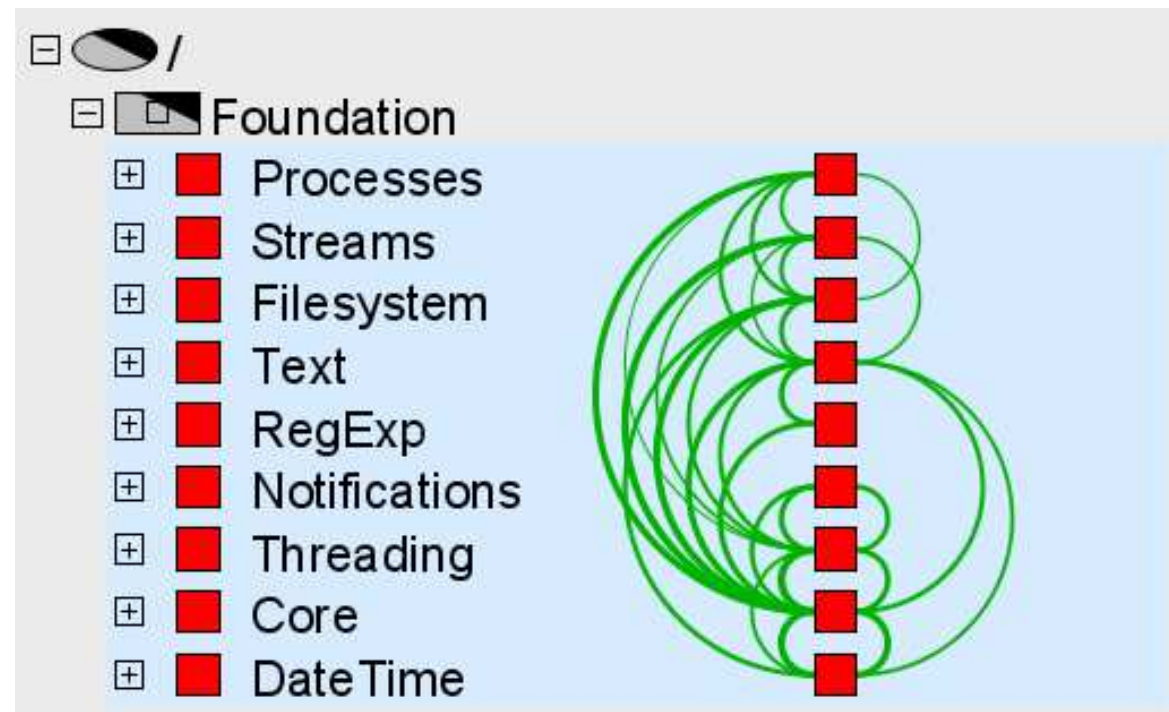
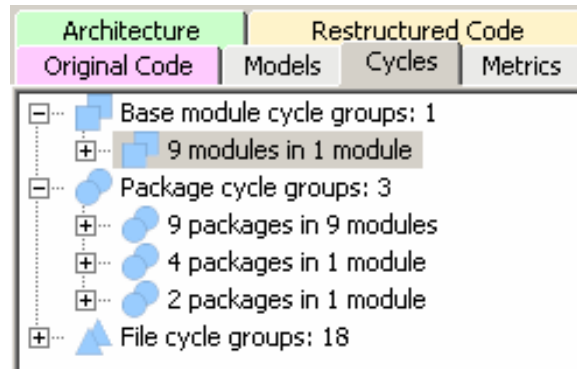


ARTIE | PHOTOGRAPHY

POCO : Foundation Dependencies



POCO : Foundation Cycles



Some erosion...

SICK





FlightGear: Architectural Analysis



Sotoarc M3.5 - flightgear20090420 @ localhost - Default* - Advanced Mode

Sotoarc Projects Model Set Options Extras Help

Architecture Restructured Code
Original Code Models Cycles Metrics

Do Not Color Nodes Show All References Undo/Redo (Modeling)

X 1(Base)

Architecture Modeling View

PACKAGE cycle groups: 2

- 21 packages
 - AIModel - src
 - ATC - src
 - ATCDCL - src
 - Aircraft - src
 - Airports - src
 - Cockpit - src
 - Environment - src
 - FDM - src
 - GUI - src
 - HUD - src/Instrumentation
 - Instrumentation - src
 - KLN89 - src/Instrumentation
 - Model - src
 - MultiPlayer - src
 - Nav aids - src
 - Network - src
 - Scenery - src
 - Scripting - src
 - Sound - src
 - Traffic - src
 - built_in - src/Cockpit
- 2 packages
- File cycle groups: 8

PACKAGES (directories)

- src
 - Instrumentation
 - KLN89
 - <files in 'Instrumentation'>
 - HUD
 - Sound
 - ATCDCL
 - Scripting
 - Environment
 - Model
 - Cockpit
 - built_in
 - <files in 'Cockpit'>
 - Scenery
 - GUI
 - Traffic
 - AIModel
 - MultiPlayer
 - Nav aids
 - Airports
 - ATC
 - Aircraft
 - Network
 - FDM
 - <files in 'FDM'>



My Podcast at se-radio.net:

<http://se-radio.net/podcast/2008-10/episode-115-architecture-analysis>

Thanks to:

- Axivion/Bauhaus:
 - Thomas Eisenbarth
 - Bernhard Berger
- Headways/Structure101:
 - Ian Sutton
 - Paul Hickey
- Programming Research:
 - Richard Corden
 - Fergus Bolger
- Software Tomography: SotoArc
 - Thomas Schoen
 - Heinrich Rust