Declarative Meta Programming

Andy Kellens (akellens@vub.ac.be)

In this lecture ...

- Goal of Declarative Meta Programming
- Meta Programming
- SOUL
- Overview of the predicates in LiCoR
- Examples
- Detecting the composite design pattern
- Applications

A bit of context

- Ongoing research at VUB and UCL
- Started in 1996 (Wuyts)
- Group of about 10 people
- Common research artefact: the SOUL language
- But different goals:
  - verifying architecture/design
  - program generation
  - declarative user interface specification
  - aspect-oriented language design
- **Common problem: supporting software evolution**

A bit of context

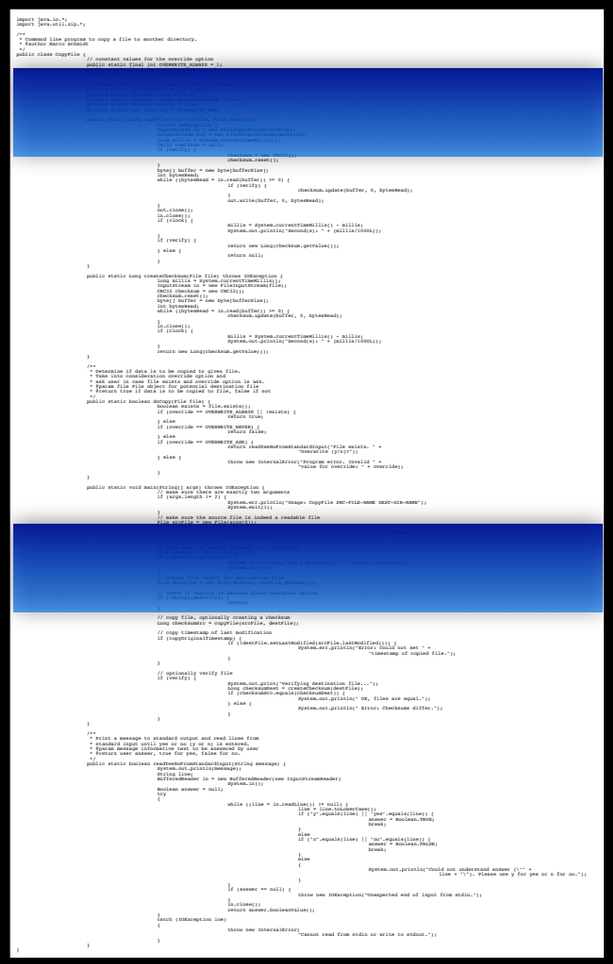
- Ongoing research at VUB and UCL
- Started in 1996 (Wuyts)
- Group of about 10 people
- Common research artefact: the SOUL language
- But different goals:
  - verifying architecture/design
  - program generation
  - declarative user interface specification
  - aspect-oriented language design
- Common problem: supporting software evolution

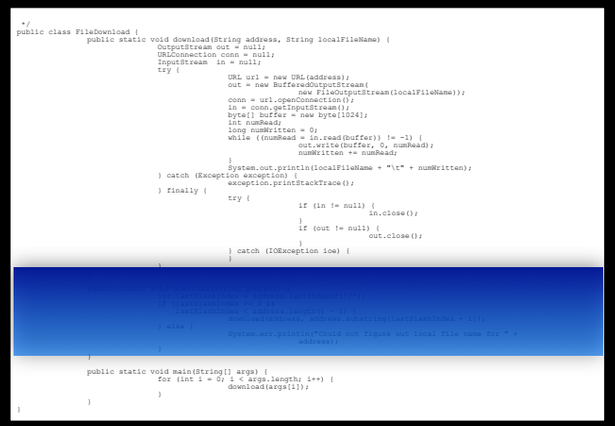
Documenting software ...

```java
import java.io.*;
import java.util.zip.*;

/**
 * Command line program to copy a file to another directory.
 * @author Marco Schmidt
 */
public class CopyFile {
		// constant values for the override option
		public static final int OVERWRITE_ALWAYS = 1;
		public static final int OVERWRITE_NEVER = 2;
		public static final int OVERWRITE_ASK = 3;

		// program options initialized to default values
		private static int bufferSize = 4 * 1024;
		private static boolean clock = true;
		private static boolean copyOriginalTimestamp = true;
		private static boolean verify = true;
		private static int override = OVERWRITE_ASK;

		public static Long copyFile(File srcFile, File destFile)
				throws IOException {
				InputStream in = new FileInputStream(srcFile);
				OutputStream out = new FileOutputStream(destFile);
				long millis = System.currentTimeMillis();
				CRC32 checksum = null;
				if (verify) {
						checksum = new CRC32();
						checksum.reset();
				}
				byte[] buffer = new byte[bufferSize];
				int bytesRead;
				while ((bytesRead = in.read(buffer)) >= 0) {
						if (verify) {
								checksum.update(buffer, 0, bytesRead);
						}
						out.write(buffer, 0, bytesRead);
				}
				out.close();
				in.close();
				if (clock) {
						millis = System.currentTimeMillis() - millis;
						System.out.println("Second(s): " + (millis/1000L));
				}
				if (verify) {
						return new Long(checksum.getValue());
				} else {
						return null;
				}
		}

		public static Long createChecksum(File file) throws IOException {
				long millis = System.currentTimeMillis();
				InputStream in = new FileInputStream(file);
				CRC32 checksum = new CRC32();
				checksum.reset();
				byte[] buffer = new byte[bufferSize];
				int bytesRead;
				while ((bytesRead = in.read(buffer)) >= 0) {
						checksum.update(buffer, 0, bytesRead);
				}
				in.close();
				if (clock) {
						millis = System.currentTimeMillis() - millis;
						System.out.println("Second(s): " + (millis/1000L));
				}
				return new Long(checksum.getValue());
		}

		/**
		 * Determine if data is to be copied to given file.
		 * Take into consideration override option and
		 * ask user in case file exists and override option is ask.
		 * @param file File object for potential destination file
		 * @return true if data is to be copied to file, false if not
		 */
		public static boolean doCopy(File file) {
				boolean exists = file.exists();
				if (override == OVERWRITE_ALWAYS || !exists) {
						return true;
				} else
				if (override == OVERWRITE_NEVER) {
						return false;
				} else
				if (override == OVERWRITE_ASK) {
						return readYesNoFromStandardInput("File exists. " +
								"Overwrite (y/n)?");
				} else {
						throw new InternalError("Program error. Invalid " +
								"value for override: " + override);
				}
		}

		public static void main(String[] args) throws IOException {
				// make sure there are exactly two arguments
				if (args.length != 2) {
						System.err.println("Usage: CopyFile SRC-FILE-NAME DEST-DIR-NAME");
						System.exit(1);
				}
				// make sure the source file is indeed a readable file
				File srcFile = new File(args[0]);
				if (!srcFile.isFile() || !srcFile.canRead()) {
						System.err.println("Not a readable file: " + srcFile.getName());
						System.exit(1);
				}
				// make sure the second argument is a directory
				File destDir = new File(args[1]);
				if (!destDir.isDirectory()) {
						System.err.println("Not a directory: " + destDir.getName());
						System.exit(1);
				}
				// create File object for destination file
				File destFile = new File(destDir, srcFile.getName());

				// check if copying is desired given overwrite option
				if (!doCopy(destFile)) {
						return;
				}

				// copy file, optionally creating a checksum
				Long checksumSrc = copyFile(srcFile, destFile);

				// copy timestamp of last modification
				if (copyOriginalTimestamp) {
						if (!destFile.setLastModified(srcFile.lastModified())) {
								System.err.println("Error: Could not set " +
										"timestamp of copied file.");
						}
				}

				// optionally verify file
				if (verify) {
						System.out.print("Verifying destination file...");
						Long checksumDest = createChecksum(destFile);
						if (checksumSrc.equals(checksumDest)) {
								System.out.println(" OK, files are equal.");
						} else {
								System.out.println(" Error: Checksums differ.");
						}
				}
		}

		/**
		 * Print a message to standard output and read lines from
		 * standard input until yes or no (y or n) is entered.
		 * @param message informative text to be answered by user
		 * @return user answer, true for yes, false for no.
		 */
		public static boolean readYesNoFromStandardInput(String message) {
				System.out.println(message);
				String line;
				BufferedReader in = new BufferedReader(new InputStreamReader(
						System.in));
				Boolean answer = null;
				try
				{
						while ((line = in.readLine()) != null) {
								line = line.toLowerCase();
								if ("y".equals(line) || "yes".equals(line)) {
										answer = Boolean.TRUE;
										break;
								}
								else
								if ("n".equals(line) || "no".equals(line)) {
										answer = Boolean.FALSE;
										break;
								}
								else
								{
										System.out.println("Could not understand answer (\"" +
												line + "\"). Please use y for yes or n for no.");
								}
						}
						if (answer == null) {
								throw new IOException("Unexpected end of input from stdin.");
						}
						in.close();
						return answer.booleanValue();
				}
				catch (IOException ioe)
				{
						throw new InternalError(
								"Cannot read from stdin or write to stdout.");
				}
		}
}
```

```java
 */
public class FileDownload {
		public static void download(String address, String localFileName) {
				OutputStream out = null;
				URLConnection conn = null;
				InputStream  in = null;
				try {
						URL url = new URL(address);
						out = new BufferedOutputStream(
								new FileOutputStream(localFileName));
						conn = url.openConnection();
						in = conn.getInputStream();
						byte[] buffer = new byte[1024];
						int numRead;
						long numWritten = 0;
						while ((numRead = in.read(buffer)) != -1) {
								out.write(buffer, 0, numRead);
								numWritten += numRead;
						}
						System.out.println(localFileName + "\t" + numWritten);
				} catch (Exception exception) {
						exception.printStackTrace();
				} finally {
						try {
								if (in != null) {
										in.close();
								}
								if (out != null) {
										out.close();
								}
						} catch (IOException ioe) {
						}
				}
		}

		public static void download(String address) {
				int lastSlashIndex = address.lastIndexOf('/');
				if (lastSlashIndex >= 0 &&
				    lastSlashIndex < address.length() - 1) {
						download(address, address.substring(lastSlashIndex + 1));
				} else {
						System.err.println("Could not figure out local file name for " +
								address);
				}
		}

		public static void main(String[] args) {
				for (int i = 0; i < args.length; i++) {
						download(args[i]);
				}
		}
}
```

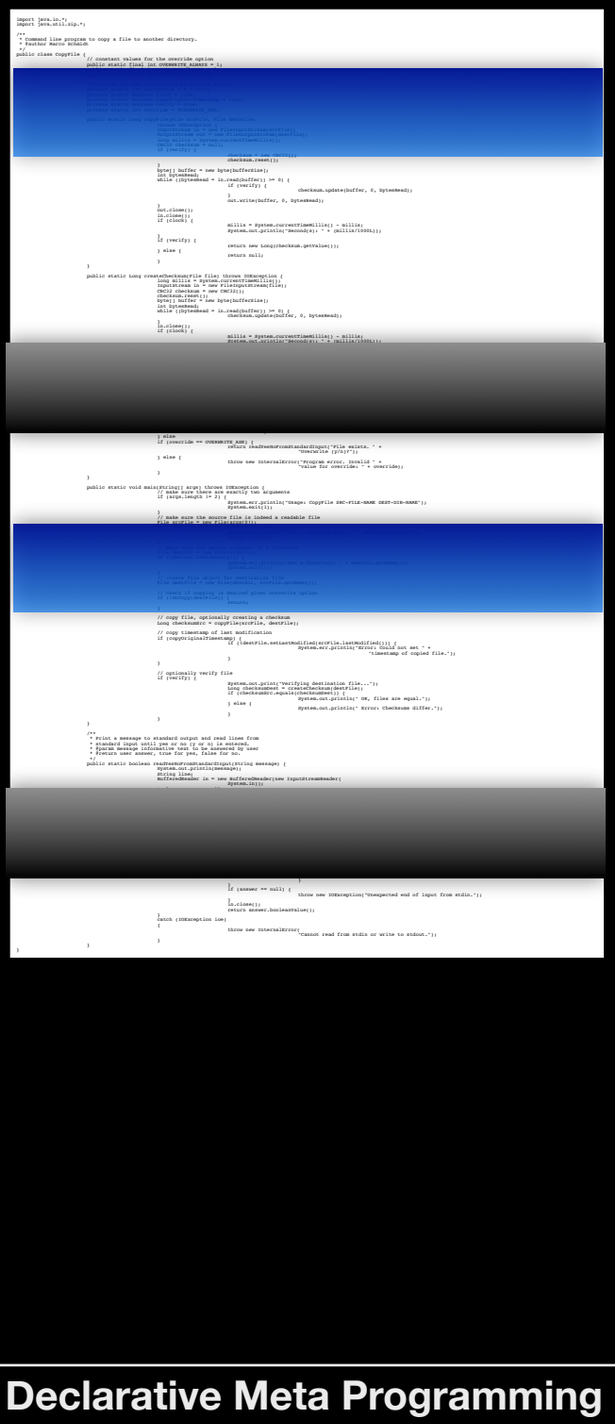
Documenting software ...

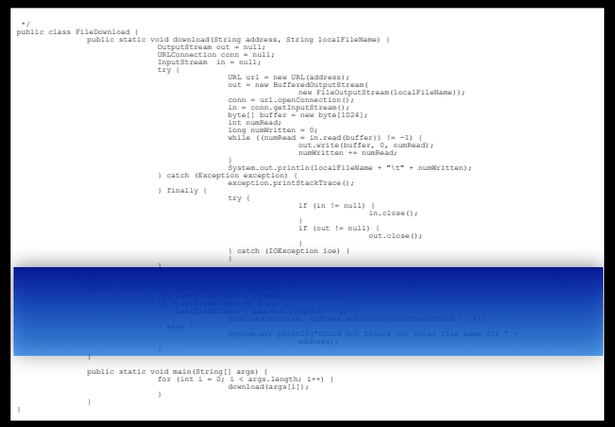




Calls to database
- after state change
- follow DB protocol

Documenting software ...

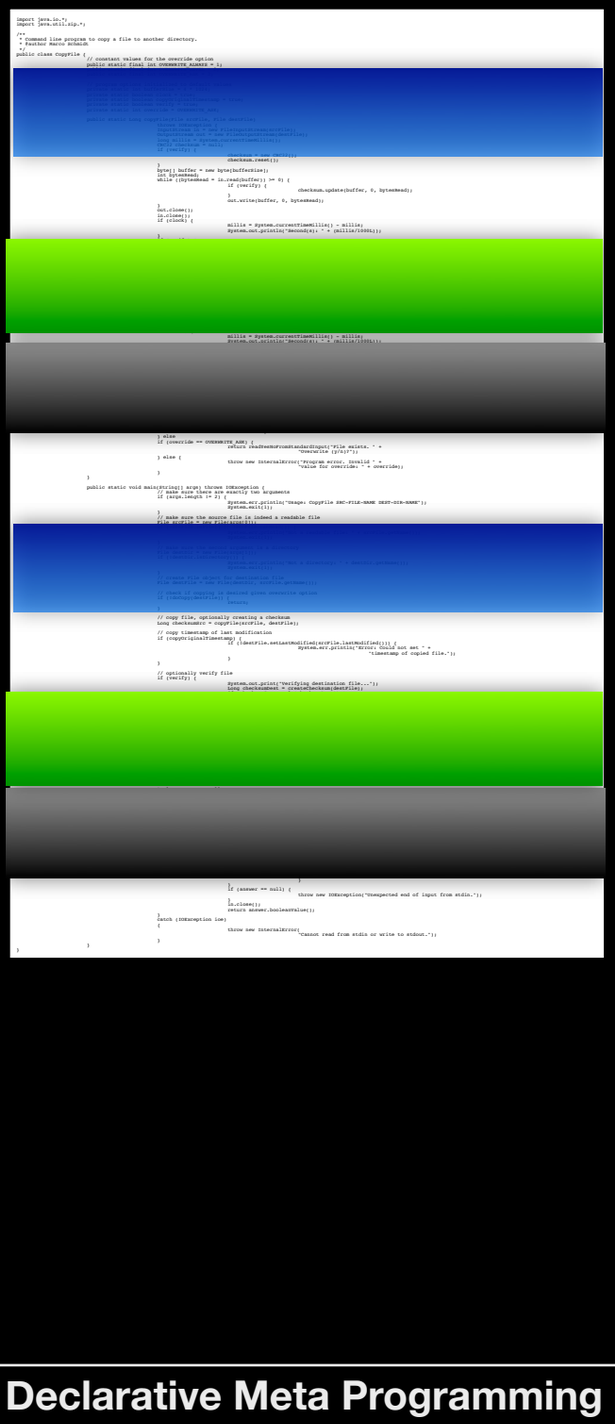


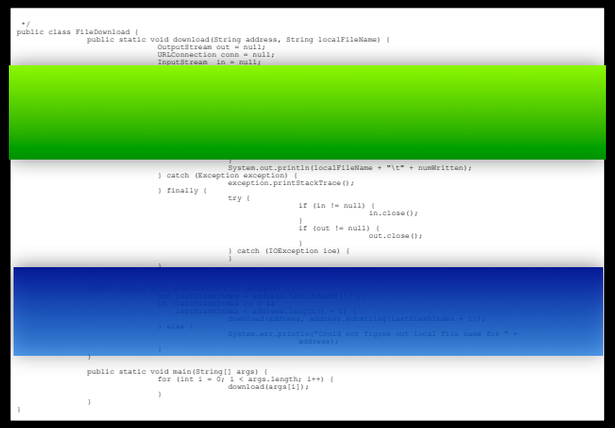


Calls to database
- after state change
- follow DB protocol

Factory methods
- create objects consistently

Documenting software ...





Calls to database

- after state change
- follow DB protocol

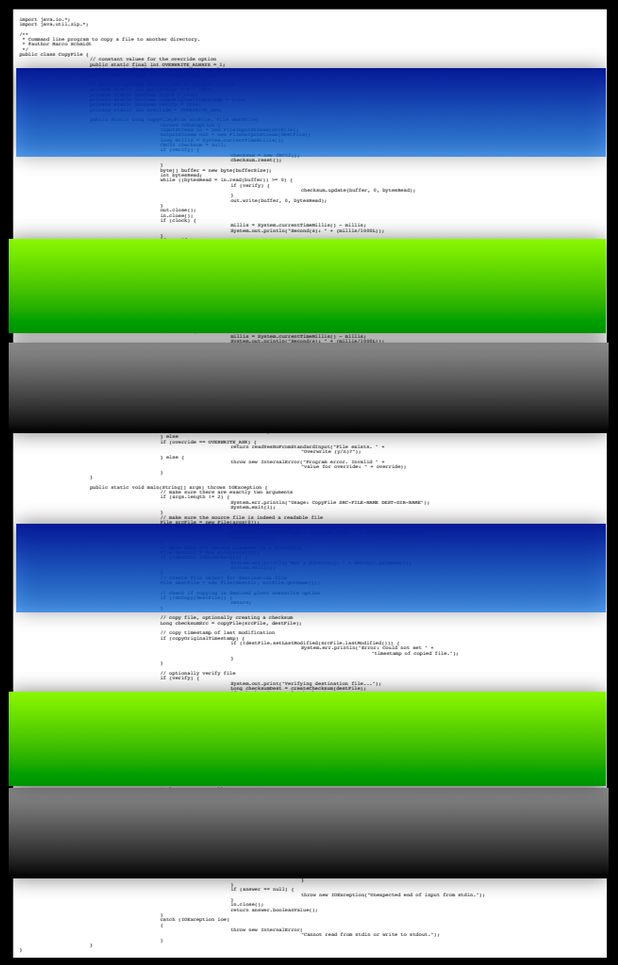
Factory methods

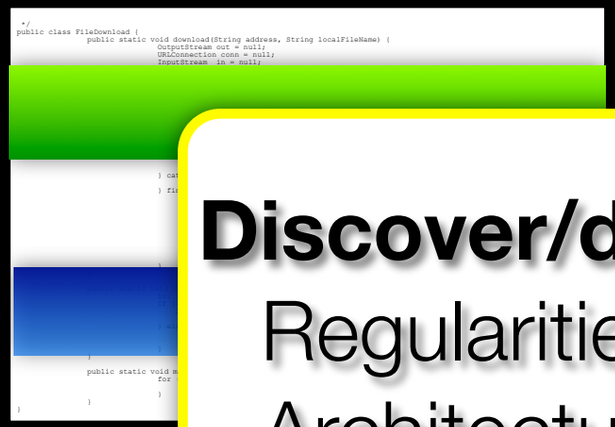
- create objects consistently

Coding conventions

- e.g. all events prefixed Event*
- follow for consistency

Documenting software ...





Discover/document:

- Regularities
- Architecture
- Idioms
- Patterns
- Conventions

- Bugs
- Bad Smells

Calls to database

er state change

ow DB protocol

ctory methods

ate objects

istently

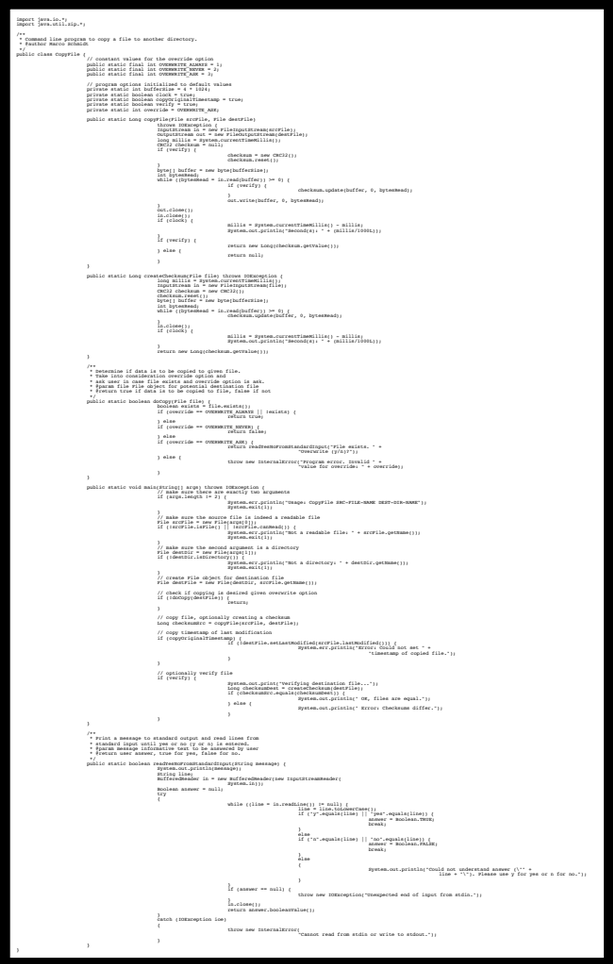
ding conventions

. all events prefixed

t*

ow for consistency

Evolution ...



```
import java.io.*;
import java.util.zip.*;

/**
 * Command line program to copy a file to another directory.
 * Author Marco ROMANO
 */
public class CopyFile {
                        // constant values for the override option
                        public static final int OVERWRITE_ALWAYS = 1;
                        public static final int OVERWRITE_NEVER = 2;
                        public static final int OVERWRITE_ASK = 3;

                        // program options initialized to default values
                        private static int buffersize = 4 * 1024;
                        private static boolean clock = true;
                        private static boolean copyOrigLastTimestamp = true;
                        private static boolean verify = true;
                        private static int overwrite = OVERWRITE_ASK;

                        public static Long copyFile(File srcFile, File destFile)
                                        throws IOException {
                                        InputStream in = new FileInputStream(srcFile);
                                        OutputStream out = new FileOutputStream(destFile);
                                        long millis = System.currentTimeMillis();
                                        CRC32 checksum = null;
                                        if (verify) {
                                                        checksum = new CRC32();
                                                        checksum.reset();
                                        }
                                        byte[] buffer = new byte[buffersize];
                                        int bytesRead;
                                        while ((bytesRead = in.read(buffer)) >= 0) {
                                                        if (verify) {
                                                                        checksum.update(buffer, 0, bytesRead);
                                                        }
                                                        out.write(buffer, 0, bytesRead);
                                        }
                                        out.close();
                                        in.close();
                                        if (clock) {
                                                        millis = System.currentTimeMillis() - millis;
                                                        System.out.println("Seconds(s): " + (millis/1000L));
                                        }
                                        if (verify) {
                                                        return new Long(checksum.getValue());
                                        } else {
                                                        return null;
                                        }
                        }
                        public static Long createChecksum(File file) throws IOException {
                                        long millis = System.currentTimeMillis();
                                        InputStream in = new FileInputStream(file);
                                        CRC32 checksum = new CRC32();
                                        checksum.reset();
                                        byte[] buffer = new byte[buffersize];
                                        int bytesRead;
                                        while ((bytesRead = in.read(buffer)) >= 0) {
                                                        checksum.update(buffer, 0, bytesRead);
                                        }
                                        in.close();
                                        if (clock) {
                                                        millis = System.currentTimeMillis() - millis;
                                                        System.out.println("Seconds(s): " + (millis/1000L));
                                        }
                                        return new Long(checksum.getValue());
                        }

                        /**
                         * Determine if data is to be copied to given file.
                         * Take into consideration override option and
                         * ask user if file exists and override option is ask.
                         * @param file file object for potential destination file
                         * @return true if data is to be copied to file, false if not
                         */
                        public static boolean doCopy(File file) {
                                        boolean exists = file.exists();
                                        if (overide == OVERWRITE_ALWAYS || !exists) {
                                                        return true;
                                        } else
                                        if (overide == OVERWRITE_NEVER) {
                                                        return false;
                                        } else
                                        if (overide == OVERWRITE_ASK) {
                                                        return readYesNoFromStandardInput("File exists. " +
                                                                                        "Overwrite (y/n)?");
                                        } else {
                                                        throw new InternalError("Program error. Invalid " +
                                                                                        "value for overide: " + overide);
                                        }
                        }
                        public static void main(String[] args) throws IOException {
                                        // make sure there are exactly two arguments
                                        if (args.length != 2) {
                                                        System.err.println("Usage: CopyFile SRC-FILE-NAME DEST-DIR-NAME");
                                                        System.exit(1);
                                        }
                                        // make sure the source file is indeed a readable file
                                        File srcFile = new File(args[0]);
                                        if (!srcFile.isFile() || !srcFile.canRead()) {
                                                        System.err.println("Not a readable file: " + srcFile.getName());
                                                        System.exit(1);
                                        }
                                        // make sure the second argument is a directory
                                        File destDir = new File(args[1]);
                                        if (!destDir.isDirectory()) {
                                                        System.err.println("Not a directory: " + destDir.getName());
                                                        System.exit(1);
                                        }
                                        // create File object for destination file
                                        File destFile = new File(destDir, srcFile.getName());

                                        // check if copying is desired given overwrite option
                                        if (!doCopy(destFile)) {
                                                        return;
                                        }
                                        // copy file, optionally creating a checksum
                                        Long checksumSrc = copyFile(srcFile, destFile);

                                        // copy timestamp of last modification
                                        if (copyOrigLastTimestamp) {
                                                        if (!destFile.setLastModified(srcFile.lastModified())) {
                                                                        System.err.println("Error: Could not set " +
                                                                                        "timestamp of copied file.");
                                                        }
                                        }
                                        // optionally verify file
                                        if (verify) {
                                                        System.out.print("Verifying destination file...");
                                                        Long checksumDest = createChecksum(destFile);
                                                        if (checksumSrc.equals(checksumDest)) {
                                                                        System.out.println(" OK, files are equal.");
                                                        } else {
                                                                        System.out.println(" Error: Checksums differ.");
                                                        }
                                        }
                        }
                        /**
                         * Print a message to standard output and read lines from
                         * standard input until the user types y or n.
                         * @param message informative text to be answered by user
                         * @return user answer, true for yes, false for no.
                         */
                        public static boolean readYesNoFromStandardInput(String message) {
                                        System.out.println(message);
                                        String line;
                                        BufferedReader in = new BufferedReader(new InputStreamReader(
                                                                        System.in));
                                        Boolean answer = null;
                                        try
                                        {
                                                        while ((line = in.readLine()) != null) {
                                                                        line = line.toLowerCase();
                                                                        if ("y".equals(line) || "yes".equals(line)) {
                                                                                        answer = Boolean.TRUE;
                                                                                        break;
                                                                        }
                                                                        else
                                                                        if ("n".equals(line) || "no".equals(line)) {
                                                                                        answer = Boolean.FALSE;
                                                                                        break;
                                                                        }
                                                                        else
                                                                        {
                                                                                        System.out.println("Could not understand answer (\"" +
                                                                                                        line + "\"). Please use y for yes or n for no.");
                                                                        }
                                                        }
                                                        if (answer == null) {
                                                                        throw new IOException("Unexpected end of input from stdin.");
                                                        }
                                                        in.close();
                                                        return answer.booleanValue();
                                        }
                                        catch (IOException ioe)
                                        {
                                                        throw new InternalError(
                                                                        "Cannot read from stdin or write to stdout.");
                                        }
                        }
}
```

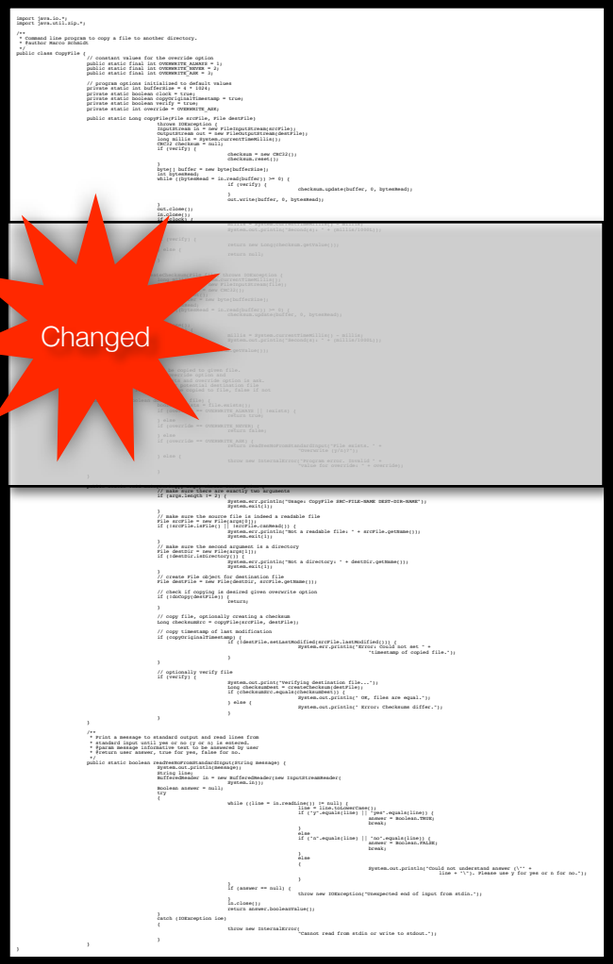
```
 */
public class FileDownload {
                        public static void download(String address, String localFileName) {
                                        OutputStream out = null;
                                        URLConnection conn = null;
                                        InputStream  in = null;
                                        try {
                                                        URL url = new URL(address);
                                                        out = new BufferedOutputStream(
                                                                        new FileOutputStream(localFileName));
                                                        conn = url.openConnection();
                                                        in = conn.getInputStream();
                                                        byte[] buffer = new byte[1024];
                                                        int numRead;
                                                        long numWritten = 0;
                                                        while ((numRead = in.read(buffer)) != -1) {
                                                                        out.write(buffer, 0, numRead);
                                                                        numWritten += numRead;
                                                        }
                                                        System.out.println(localFileName + "\t" + numWritten);
                                        } catch (Exception exception) {
                                                        exception.printStackTrace();
                                        } finally {
                                                        try {
                                                                        if (in != null) {
                                                                                        in.close();
                                                                        }
                                                                        if (out != null) {
                                                                                        out.close();
                                                                        }
                                                        } catch (IOException ioe) {
                                                        }
                                        }
                        }

                        public static void download(String address) {
                                        int lastSlashIndex = address.lastIndexOf('/');
                                        if (lastSlashIndex >= 0 &&
                                                        lastSlashIndex < address.length() - 1) {
                                                        download(address, address.substring(lastSlashIndex + 1));
                                        } else {
                                                        System.err.println("Could not figure out local file name for " +
                                                                        address);
                                        }
                        }

                        public static void main(String[] args) {
                                        for (int i = 0; i < args.length; i++) {
                                                        download(args[i]);
                                        }
                        }
}
```

Calls to database
- after state change
- follow DB protocol

Factory methods
- create objects consistently

Coding conventions
- e.g. all events prefixed Event*
- follow for consistency

Evolution ...



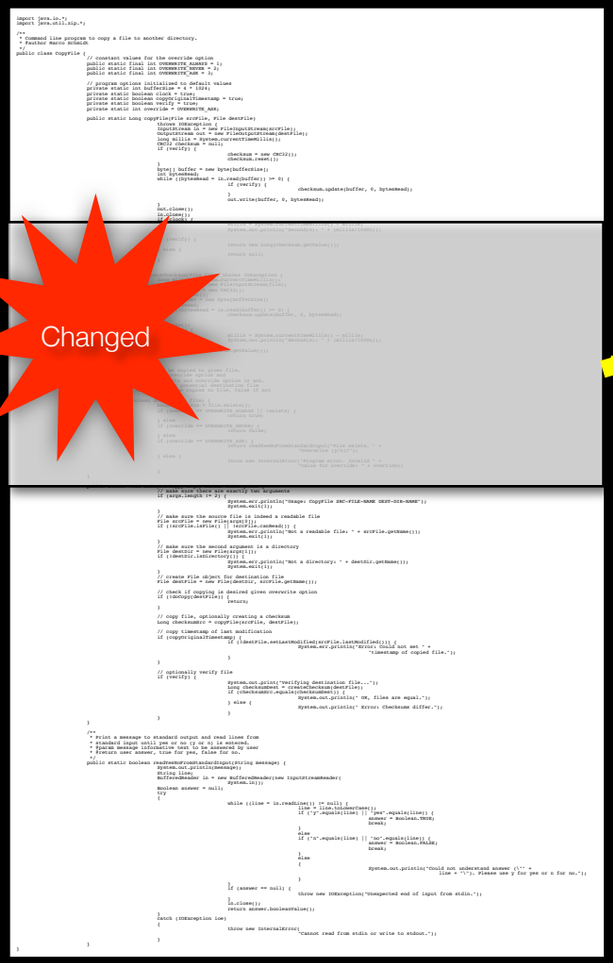
Changed

Calls to database
- after state change
- follow DB protocol

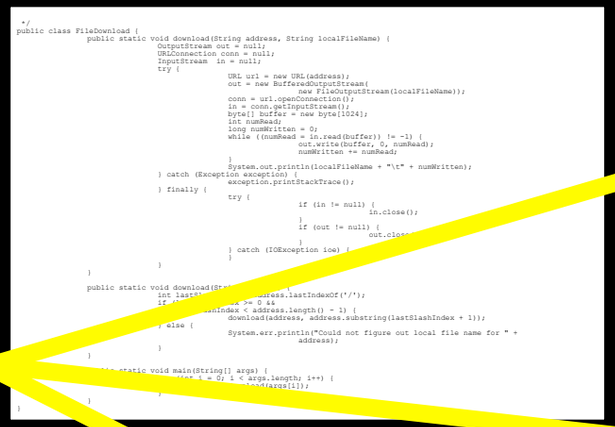
Factory methods
- create objects consistently

Coding conventions
- e.g. all events prefixed Event*
- follow for consistency

Evolution ...



Changed



???

Calls to database
- after state change
- follow DB protocol

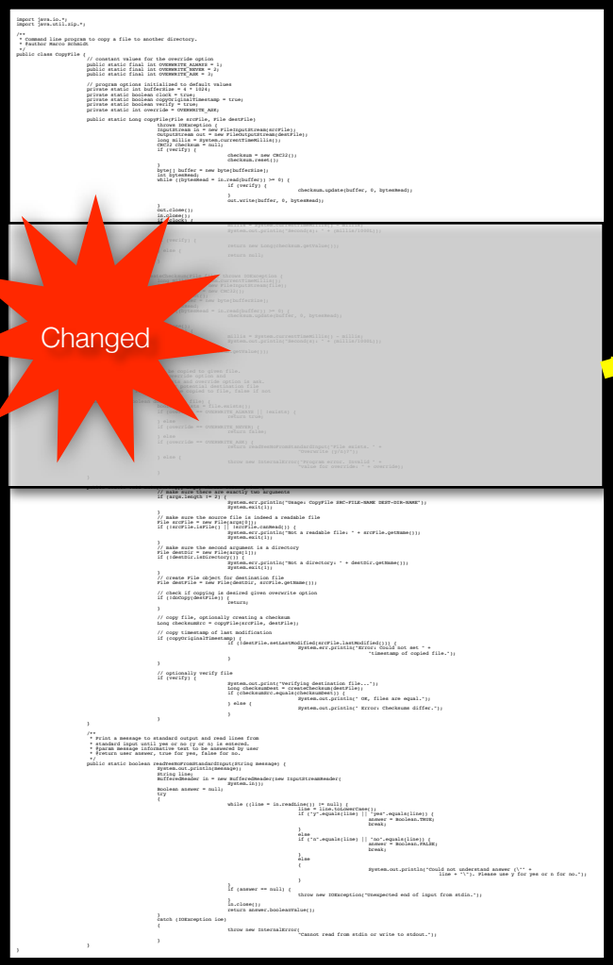
???

Factory methods
- create objects consistently

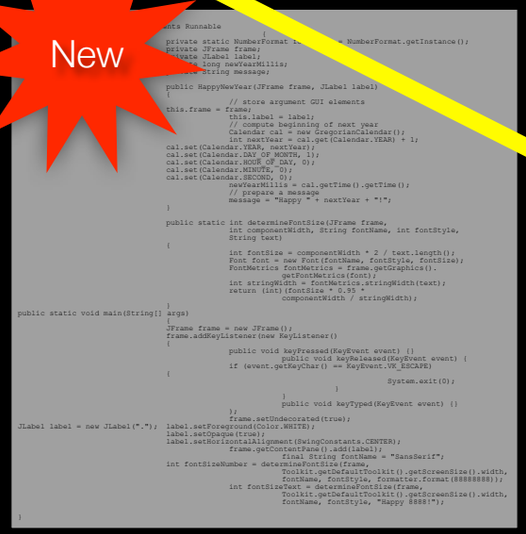
???

Coding conventions
- e.g. all events prefixed Event*
- follow for consistency

Evolution ...



Changed



New

???

???

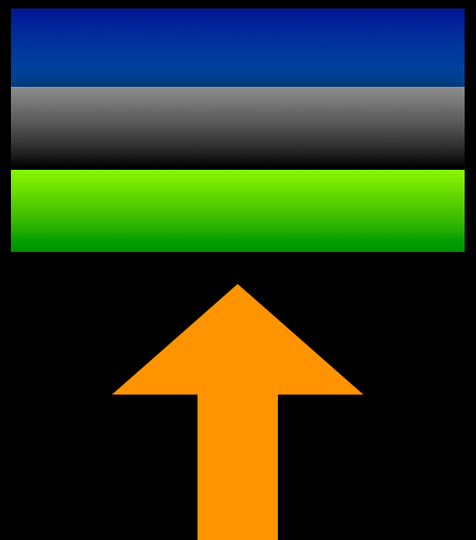
???

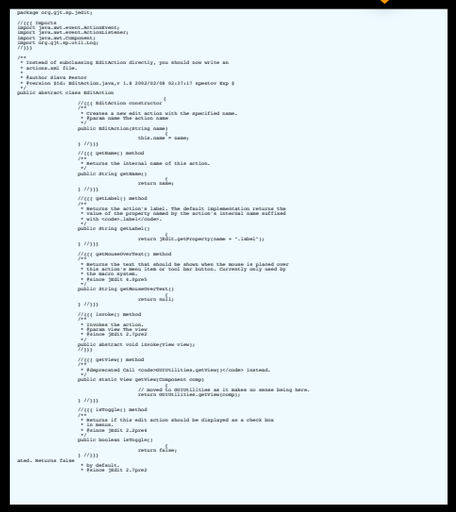
Calls to database
- after state change
- follow DB protocol

Factory methods
- create objects consistently

Coding conventions
- e.g. all events prefixed Event*
- follow for consistency

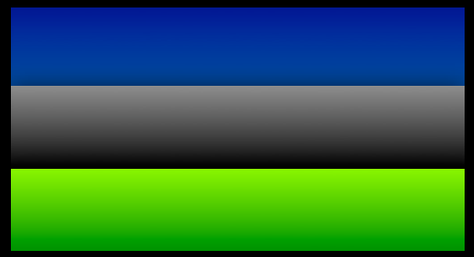
Software Erosion

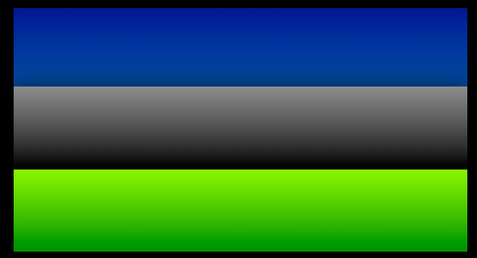


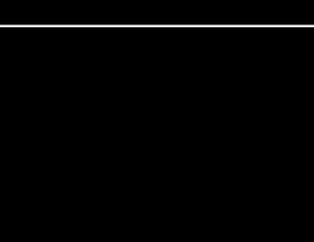


```
package org.gjt.sp.jedit;

//{{{ Imports
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.Component;
import org.gjt.sp.util.Log;
//}}}

/**
 * Instead of subclassing EditAction directly, you should now write an
 * actions.xml file.
 *
 * @author Slava Pestov
 * @version $Id: EditAction.java,v 1.8 2002/02/08 02:37:17 spestov Exp $
 */
public abstract class EditAction
{
	//{{{ EditAction constructor
	/**
	 * Creates a new edit action with the specified name.
	 * @param name The action name
	 */
	public EditAction(String name)
	{
		this.name = name;
	} //}}}

	//{{{ getName() method
	/**
	 * Returns the internal name of this action.
	 */
	public String getName()
	{
		return name;
	} //}}}

	//{{{ getLabel() method
	/**
	 * Returns the action's label. The default implementation returns the
	 * value of the property named by the action's internal name suffixed
	 * with <code>.label</code>.
	 */
	public String getLabel()
	{
		return jEdit.getProperty(name + ".label");
	} //}}}

	//{{{ getMouseOverText() method
	/**
	 * Returns the text that should be shown when the mouse is placed over
	 * a button for this item, or tool bar button. Currently only used by
	 * the macro system.
	 * @since jEdit 4.0pre3
	 */
	public String getMouseOverText()
	{
		return null;
	} //}}}

	//{{{ invoke() method
	/**
	 * Invokes the action.
	 * @param view The view
	 * @since jEdit 2.7pre2
	 */
	public abstract void invoke(View view);
	//}}}

	//{{{ getView() method
	/**
	 * @deprecated Call <code>GUIUtilities.getView()</code> instead.
	 */
	public static View getView(Component comp)
	{
		// access in GUIUtilities as it makes no sense being here.
		return GUIUtilities.getView(comp);
	} //}}}

	//{{{ isToggle() method
	/**
	 * Returns if this edit action should be displayed as a check box
	 * menu item.
	 * @since jEdit 2.7pre4
	 */
	public boolean isToggle()
	{
		return false;
	} //}}}
}
```

Software Erosion

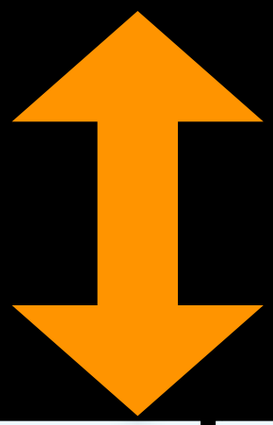


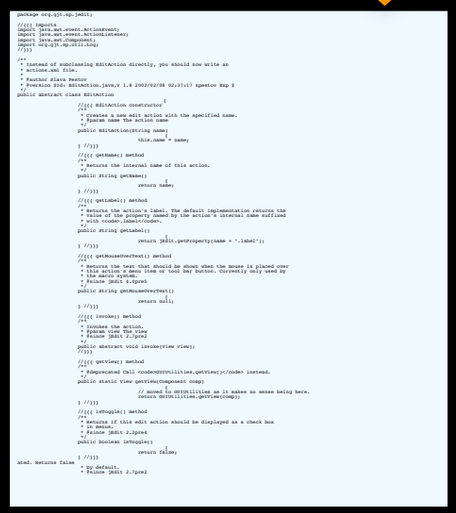


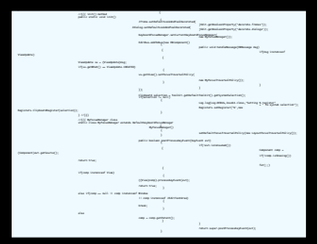


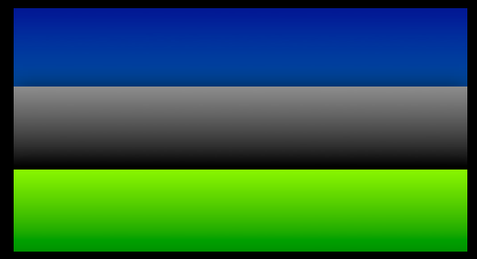
time

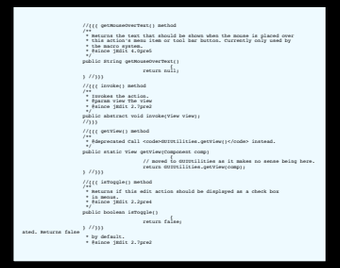
Software Erosion





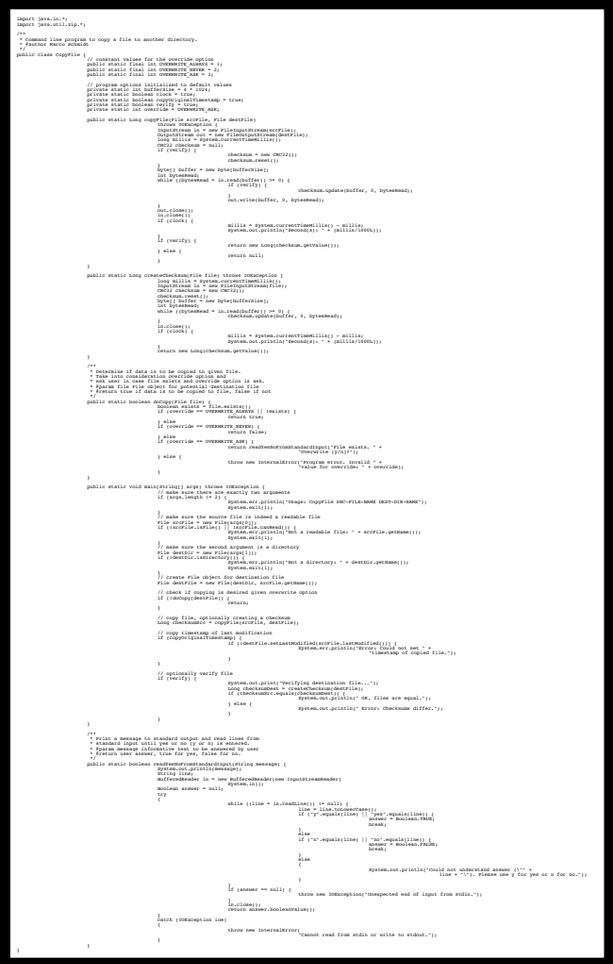


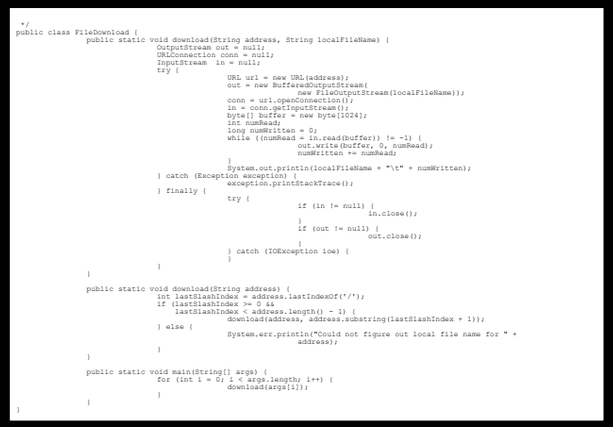




time

Small vs. large software

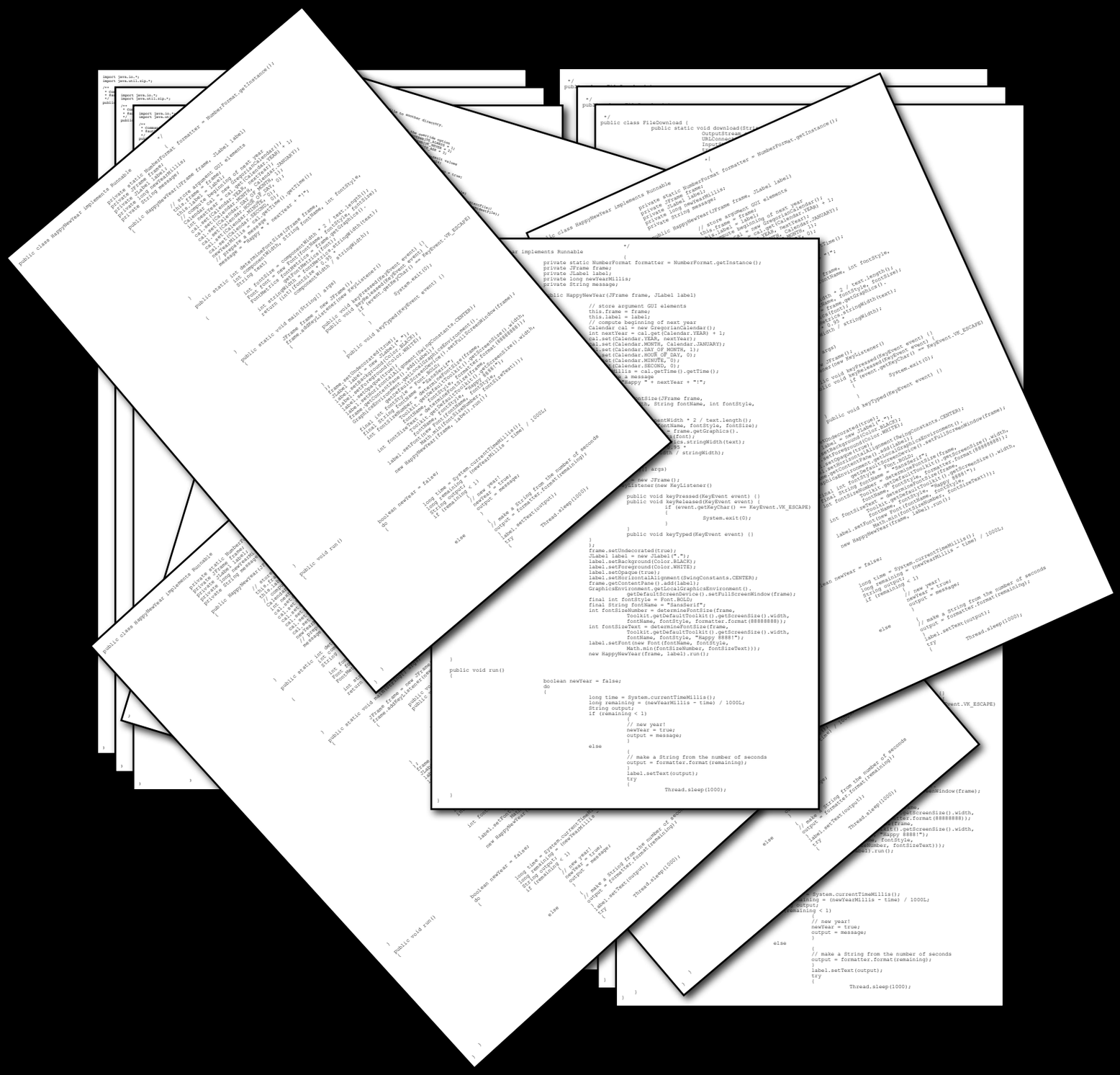




Manual does not scale

Tools needed

Small vs. large software



Manual does not scale

Tools needed

Declarative meta programming

- **Prevent software erosion**
- **Querying the source code**
  - Structure of the source code
  - Static analysis (approximate what the source code will do)
  - mix of logic programming and imperative/OO programming
- **Causal connection**
  - Keep the link with the source code
  - If code changes, queries work over new code

Meta programming

Meta-program

e.g. GCC, Eclipse, doxygen, ...

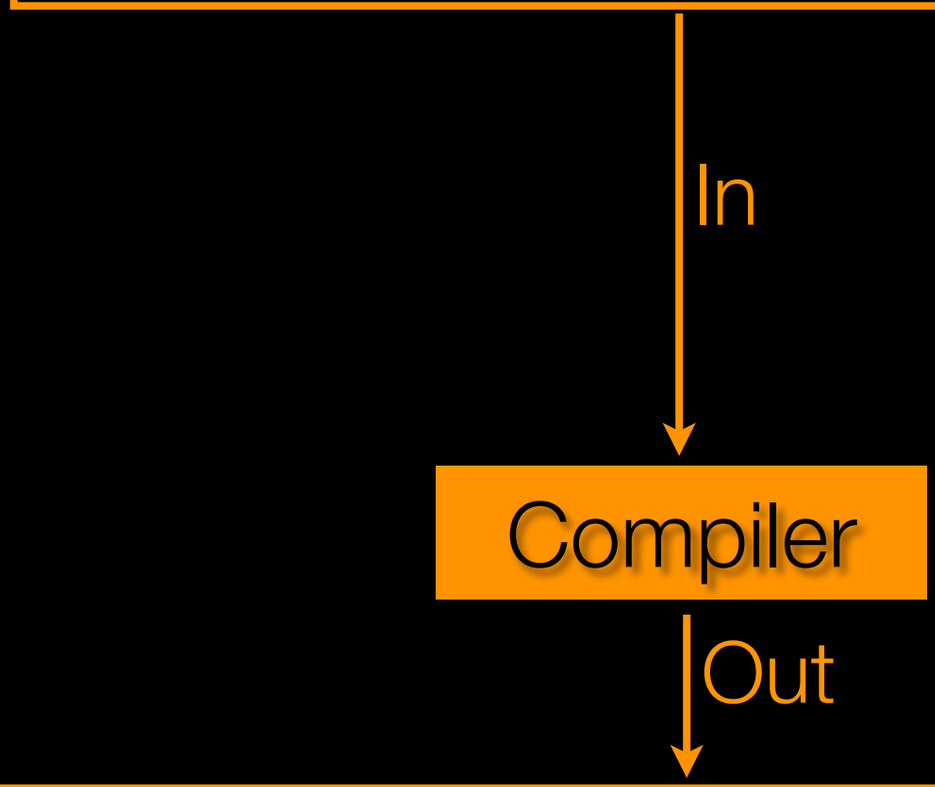
Reason and manipulate

Base language program

e.g. Smalltalk program,
Java program, ...

Example

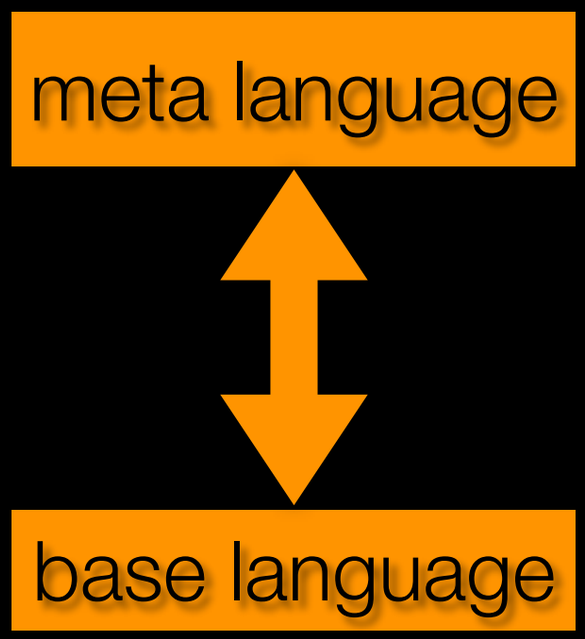
```
Transcript show: ‘Hello, World!’
```




```
short CompiledBlock numArgs=0 numTemps=0 frameSize=12

literals: ({Transcript} 'Hello, World!' )

1 <34> push Transcript
2 <1D> push 'Hello, World!'
3 <F0 80> send show:
5 <65> return
```

Meta programming language

- Programming language used to write meta programs
- Reason over/manipulate programs in *base* language

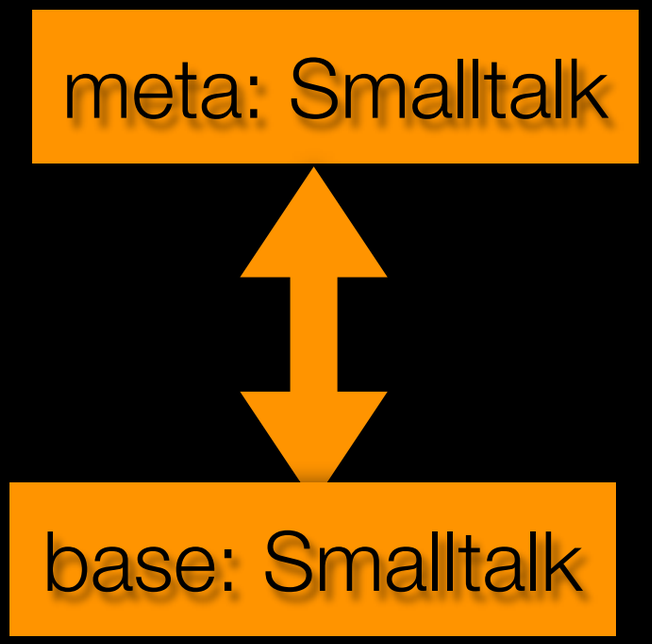


- Meta and base language can be the same

Example meta program in Smalltalk

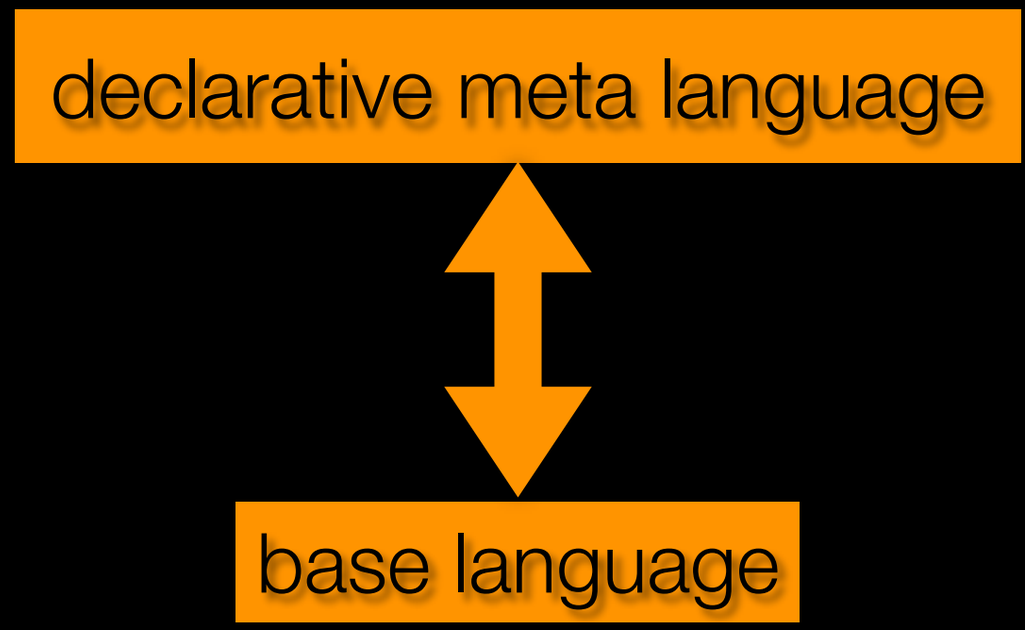
```
Smalltalk allClasses
    do:[:class |
      class selectors
           do:[:selector |
        Transcript show: class asString, '>>', selector asString; cr.
]]
```

SmalltalkMethod>>logicParseTree
SmalltalkMethod>>classesReferenced
SmalltalkMethod>>printOn:
F03_Streams_03>>codeExample
F03_Streams_03>>how
F03_Streams_03>>what
F03_Streams_03>>sortArray
F03_Streams_03>>why
F03_Streams_03>>defaultWorkspaceText
F03_Streams_03>>displayString
.....



Declarative meta programming

- **Use of declarative language as meta language**
- **In our case: Prolog**



Why Prolog?

- **As mentioned yesterday:**

  - Focus on *what*, rather than *how*

- **Source code is the fact base**
- **Use queries in order to retrieve information from that code**
- **Has proven to be very effective!**

SOUL

- Smalltalk *O*pen *U*nification *L*anguage
- Implemented on top of Smalltalk
- Prolog derivative
- Symbiosis with Smalltalk
- Libraries of predicates to reason about software

SOUL vs. Prolog

- Variables indicated by ?
- Lists using < and > instead of [and]
- *equals* predicate instead of =
- Keyword if instead of :-

```
%append(?list1, ?list2, ?list)

append(<>, ?list, ?list)
append(<?first | ?rest>, ?list2, <?first | ?list>) if
    append(?rest, ?list2, ?list).
```

Smalltalk terms

- Symbiosis with underlying Smalltalk
- Use Smalltalk objects in SOUL
- Indicated using [and]
- Use of Smalltalk values:

```
equals(?string, [‘Hello, World’])
```

- Use of Smalltalk expressions:

```
add(?number1, ?number2, [?number1 + ?number2])
```

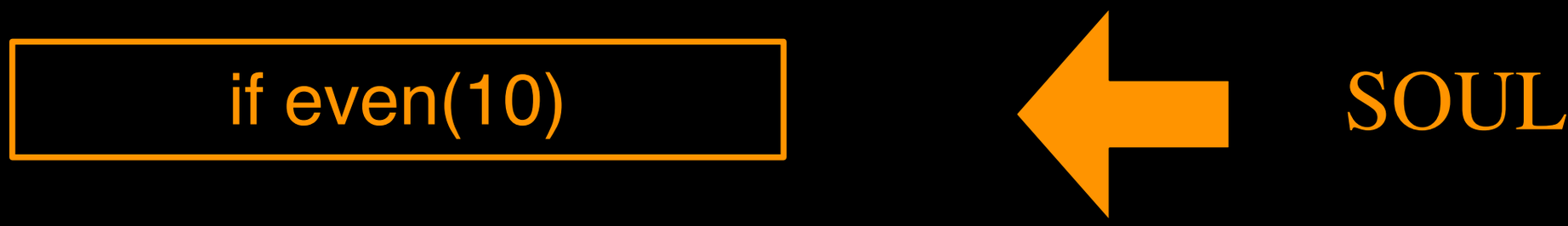
Smalltalk clauses

- Use a Smalltalk block as a logic condition
- Must evaluate to *true* or *false*
- Variables must be bound when evaluated
- Unification = evaluation

```
even(?number) if
    [(?number rem: 2) = 0]
```

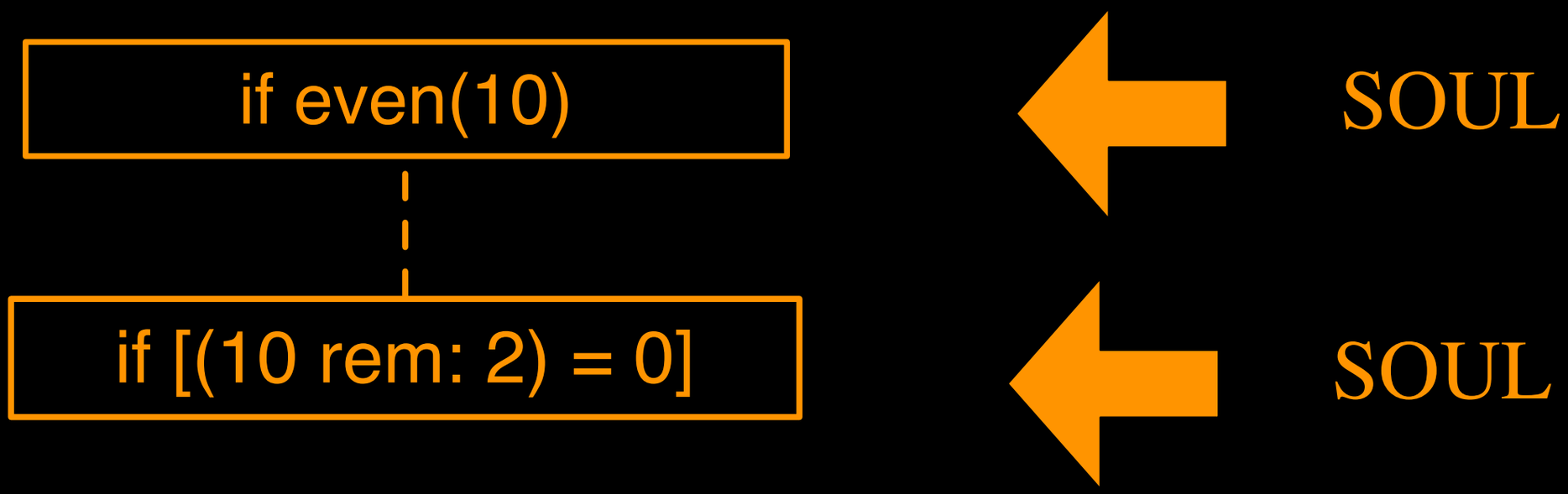
Resolution of Smalltalk clauses

```
if even(10)
```



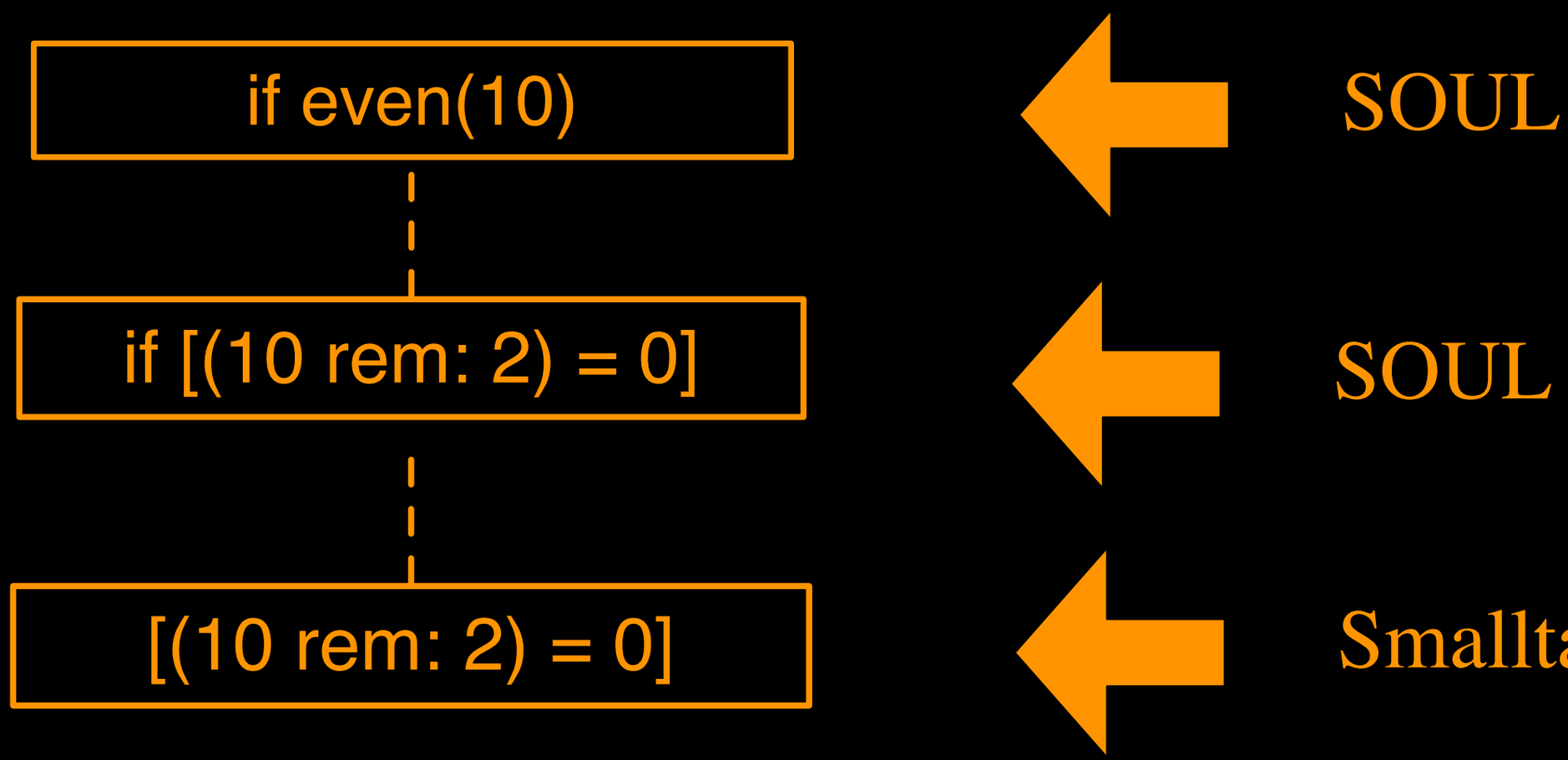
Resolution of Smalltalk clauses

```
if even(10)
```



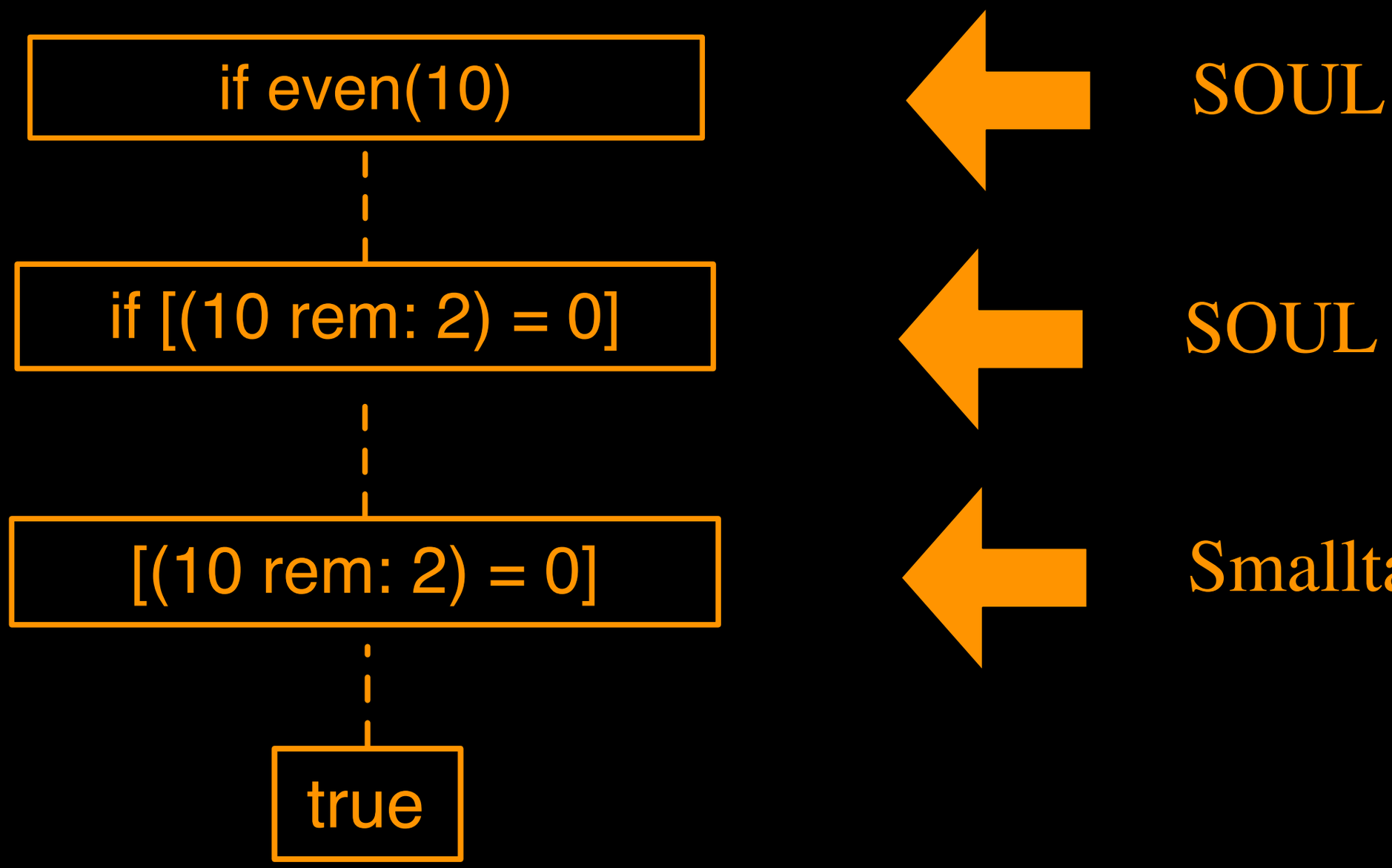
Resolution of Smalltalk clauses

```
if even(10)
```



Resolution of Smalltalk clauses

```
if even(10)
```



Example usage of symbiosis

```
if member(?number, <1,2,3>)
if member(?number, [Array with: 1 with: 2 with: 3])
```

?number -> 1
?number -> 2
?number -> 3

Smalltalk collection

```
member(?element, ?list) if
  [?list isKindOf: Collection],
  memberCollection(?element,?list)
```

SOUL list

```
member(?element, ?list) if
    list(?list),
    memberList(?element,?list)
```

Quoted code

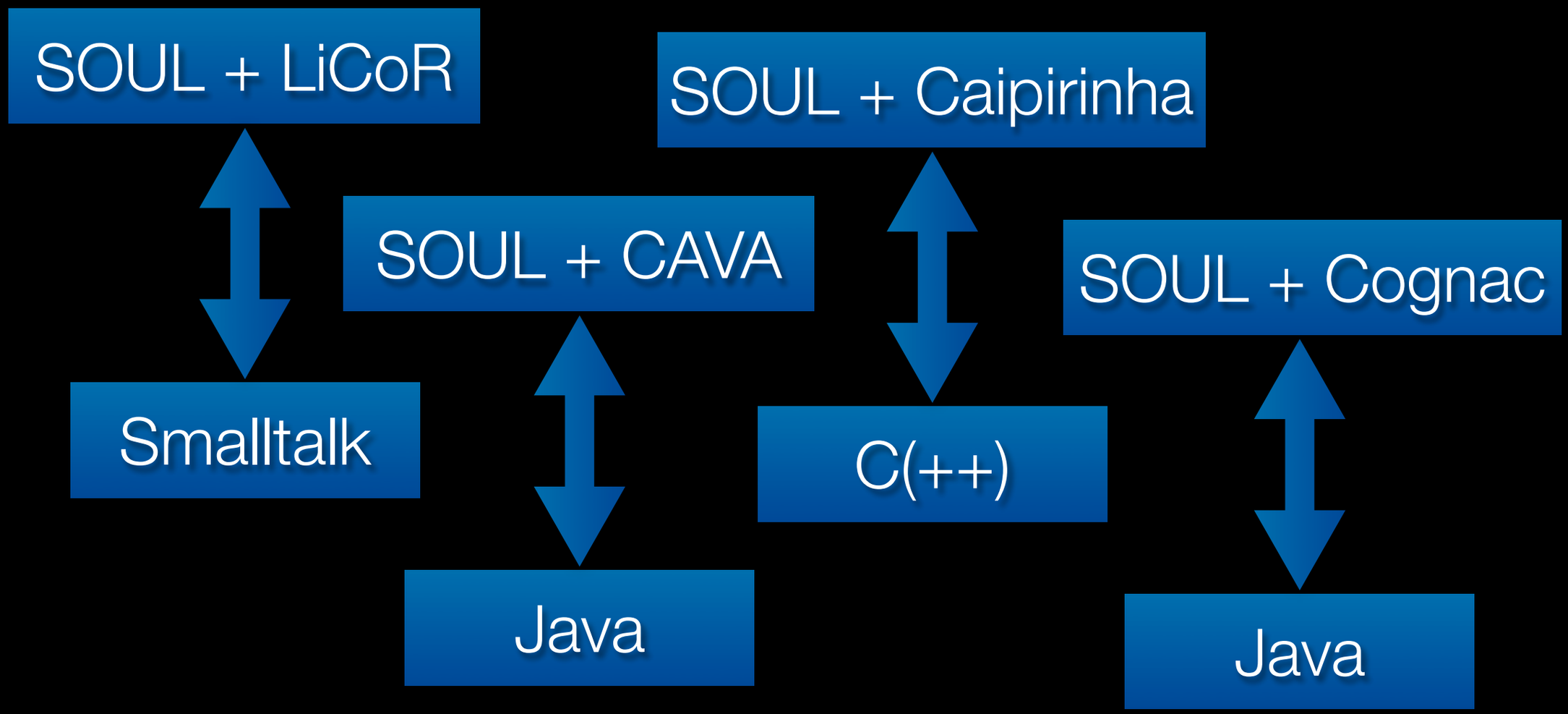
- Consider it a string with variables
- Useful for code generation
- Not evaluated by SOUL
- Indicated by { and }

```
equals(?code, {Array with: ?x with: ?y}),
equals(?x, 1),
equals(?y, 2),
equals(?array, [Compiler evaluate: ?code])
```

?code -> {Array with: 1 with: 2}
?array -> #(1 2)

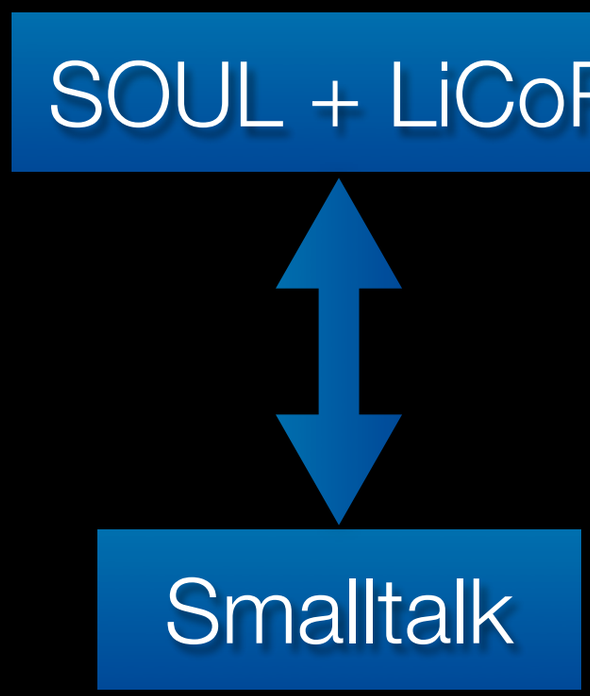
Reasoning about actual programs

- **Libraries of predicates for SOUL**
- **For Smalltalk, Java, C(++) and Cobol**

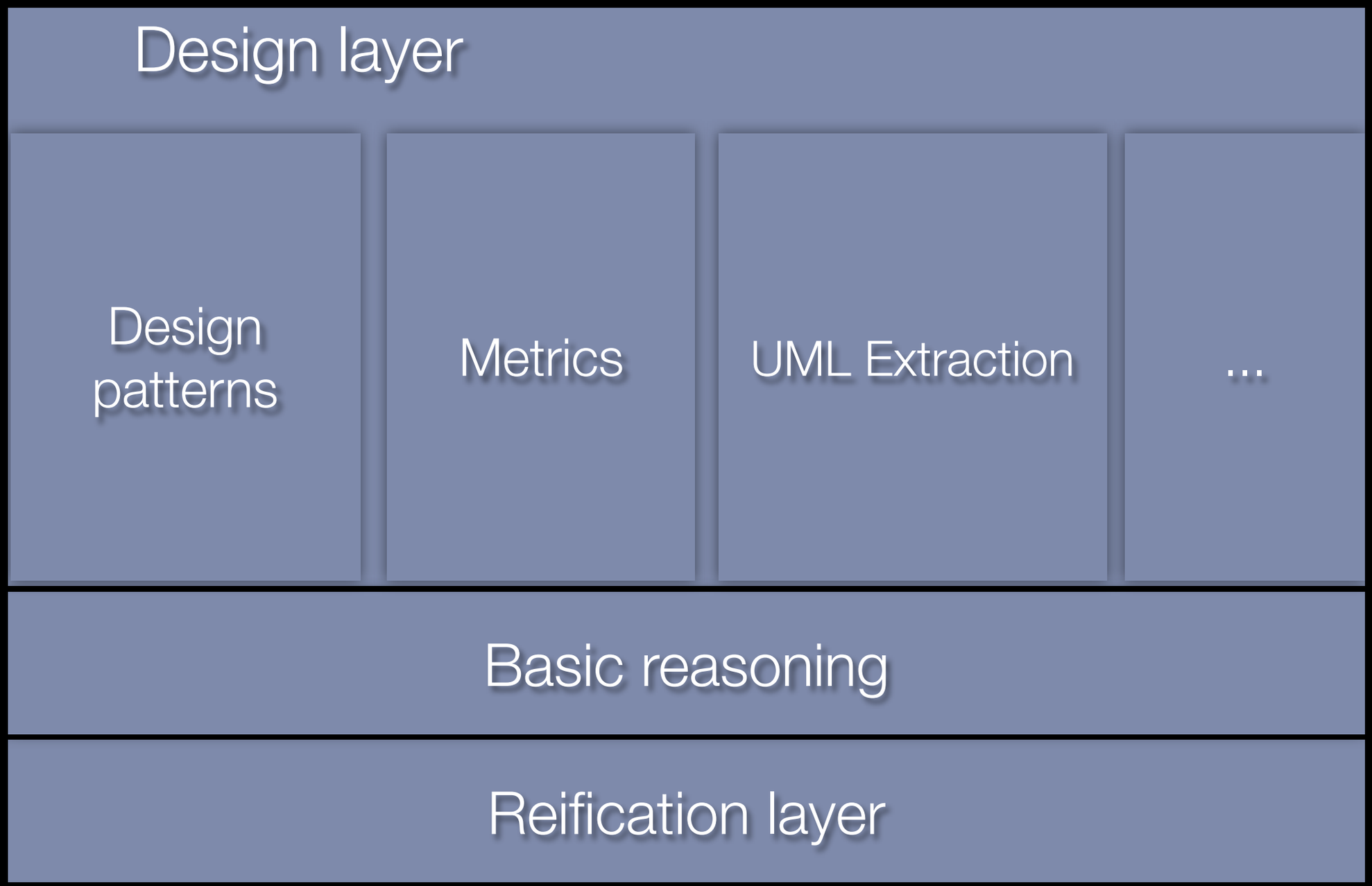


Reasoning about actual programs

- Libraries of predicates for SOUL
- For Smalltalk, Java, C(++) and Cobol



Layered design



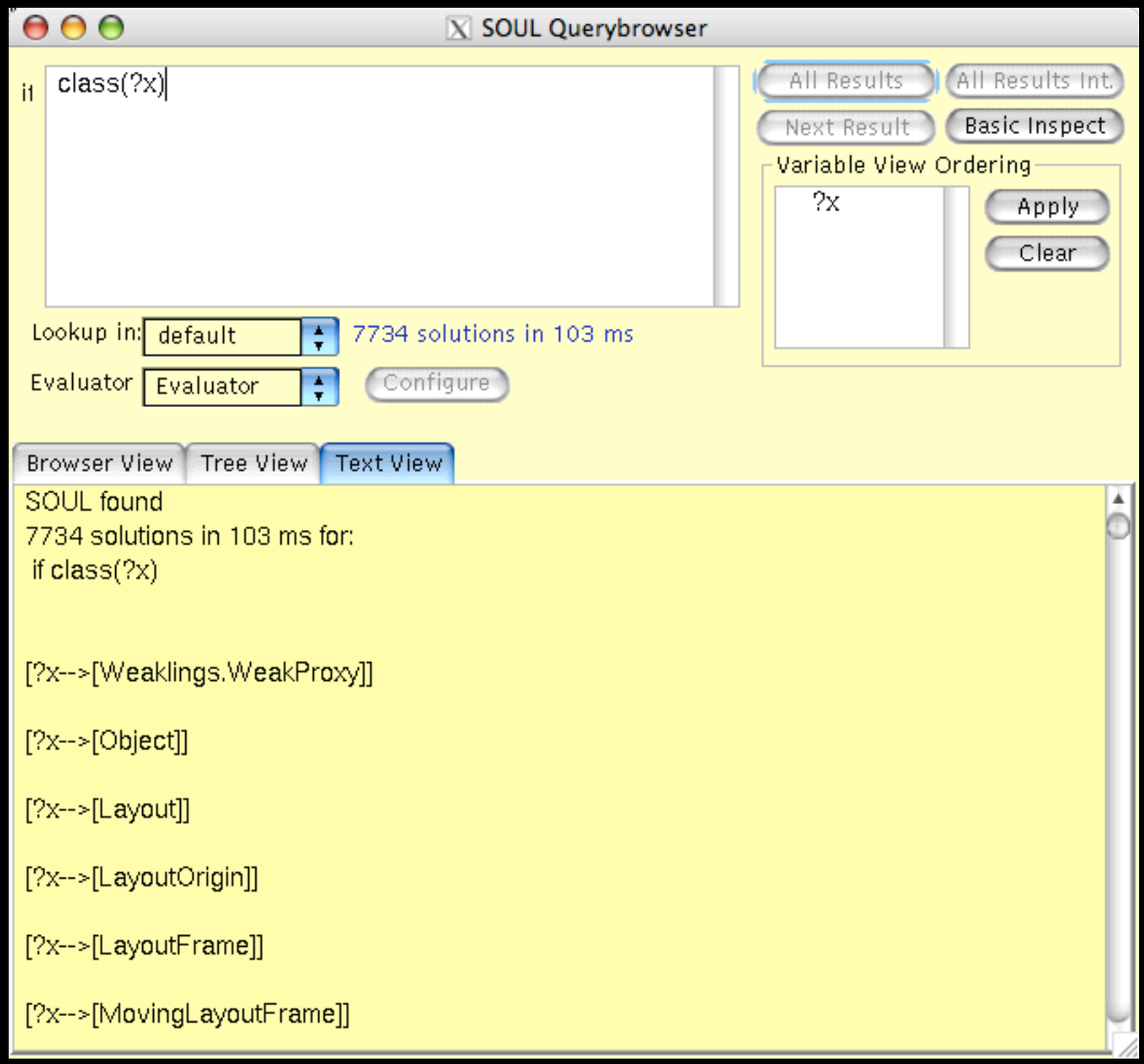
Reification layer

- **Represent program entities in SOUL**
  - E.g. classes, methods, bundles, packages, ...
- **Basic set of predicates**

```
class(?class)

method(?method)

methodInClass(?method)

classInPackage(?class, ?package)
```

Reification layer

Example


SOUL Querybrowser
if
class(?x)
All Results
All Results Int.
Next Result
Basic Inspect
Variable View Ordering
?x
Apply
Clear
Lookup in:
default
7734 solutions in 103 ms
Evaluator
Evaluator
Configure
Browser View
Tree View
Text View
SOUL found
7734 solutions in 103 ms for:
if class(?x)
[?x-->[Weaklings.WeakProxy]]
[?x-->[Object]]
[?x-->[Layout]]
[?x-->[LayoutOrigin]]
[?x-->[LayoutFrame]]
[?x-->[MovingLayoutFrame]]

Implemented using symbiosis

- **Extensive meta programming facilities in Smalltalk**
- **Use them to implement reification layer**

```
class(?class) if
   member(?class, [Smalltalk allClasses])
```

```
methodInClass(?method,?class) if
   member(?method, [?class selectors collect:[:sel |
          ?class compiledMethodAt: sel]])
```

Implemented using symbiosis

- **Extensive meta programming facilities in Smalltalk**
- **Use them to implement reification layer**

```
class(?class) if
    member(?class, [Smalltalk allClasses])
```

Link to actual implementation

Via Smalltalk

```
methodInClass(?method,?class) if
    member(?method, [?class selectors collect:[:sel |
            ?class compiledMethodAt: sel]])
```

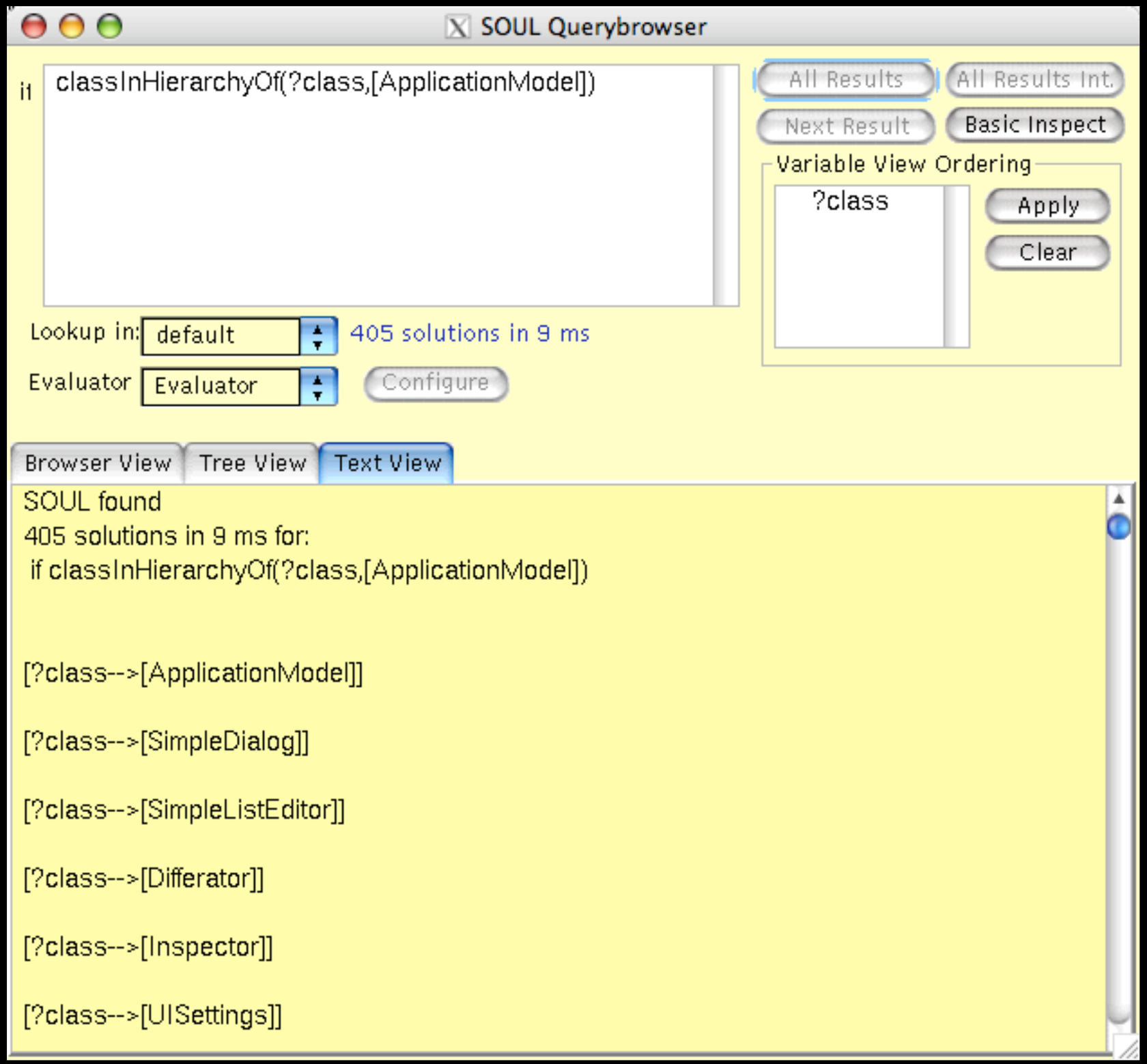
Basic reasoning

- **Built on top of reification layer**
- **Extract:**
  - basic relationships
  - abstract class predicate
  - message sends

```
classInHierarchyOf(?class, ?parent)
methodWithNameInClass(?method, ?name,?
class)
classImplements(?class,?selector)
abstractMethod(?method)
methodSendsSelector(?method,?selector)
....
```

Basic reasoning

Example


SOUL Querybrowser
if
classInHierarchyOf(?class,[ApplicationModel])
All Results
All Results Int.
Next Result
Basic Inspect
Variable View Ordering
?class
Apply
Clear
Lookup in:
default
405 solutions in 9 ms
Evaluator
Evaluator
Configure
Browser View
Tree View
Text View
SOUL found
405 solutions in 9 ms for:
if classInHierarchyOf(?class,[ApplicationModel])
[?class-->[ApplicationModel]]
[?class-->[SimpleDialog]]
[?class-->[SimpleListEditor]]
[?class-->[Differator]]
[?class-->[Inspector]]
[?class-->[UISettings]]

A more complex example

```
if methodInClass(?method, ?class),
   methodWithReturnStatement(?method,variable(?field)),
   instanceVariableInClass(?field,?class)
```

```
Person>>name
  ^name

Person>>address
  ^address
```

Accessors

SOUL Querybrowser

if

```
methodInClass(?method,?class),
methodWithReturnStatement(?method,variable(?field)),
instanceVariableInClass(?field,?class)
```

All Results | Debug | Next Result | Basic Inspect | Next x Results

Variable View Ordering

?method
?field
?class

Apply | Clear

Lookup in: LiCoR-Smalltalk — 16 solutions in 6552 ms

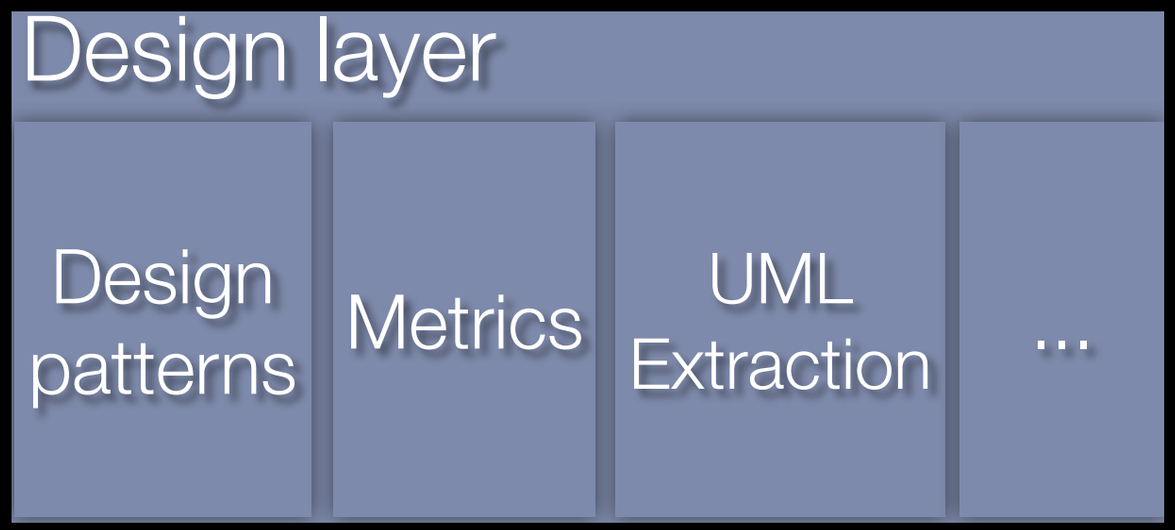
Evaluator : Evaluator — Configure

Browser View | Tree View | Text View

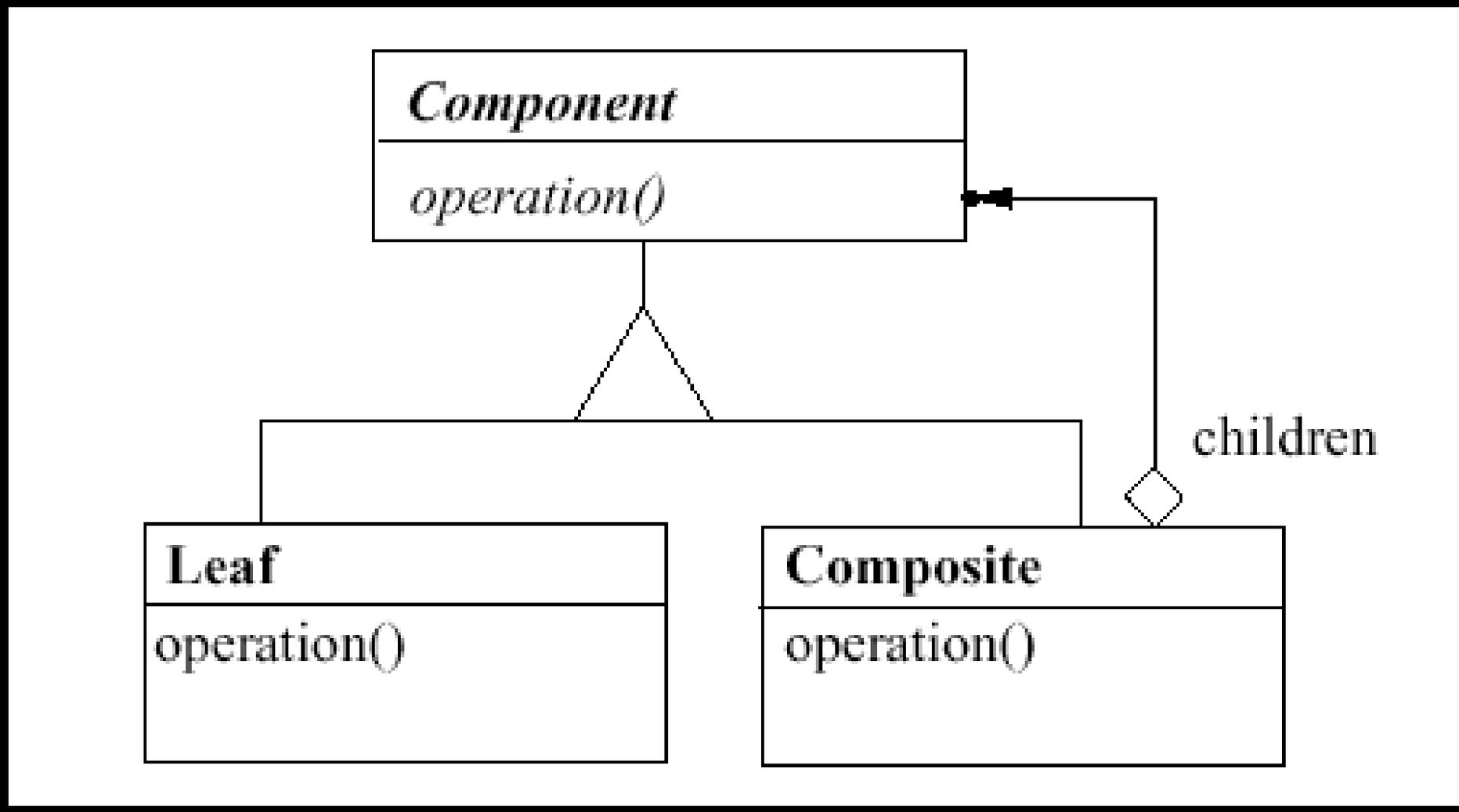
```
[?class-->[LayoutOrigin],
?field-->[#topFraction],
?method-->[LayoutOrigin>>topFraction]]

[?class-->[LayoutOrigin],
?field-->[#left],
?method-->[LayoutOrigin>>leftOffset]]

[?class-->[LayoutOrigin],
?field-->[#leftFraction],
?method-->[LayoutOrigin>>leftFraction]]

[?class-->[LayoutFrame],
?field-->[#right],
?method-->[LayoutFrame>>rightOffset]]
```

Design layer

- **Set of predicates to:**
  - detect design patterns
  - detect bad smells
  - calculate metrics
  - ...
- **Using the predicates from the underlying layers**

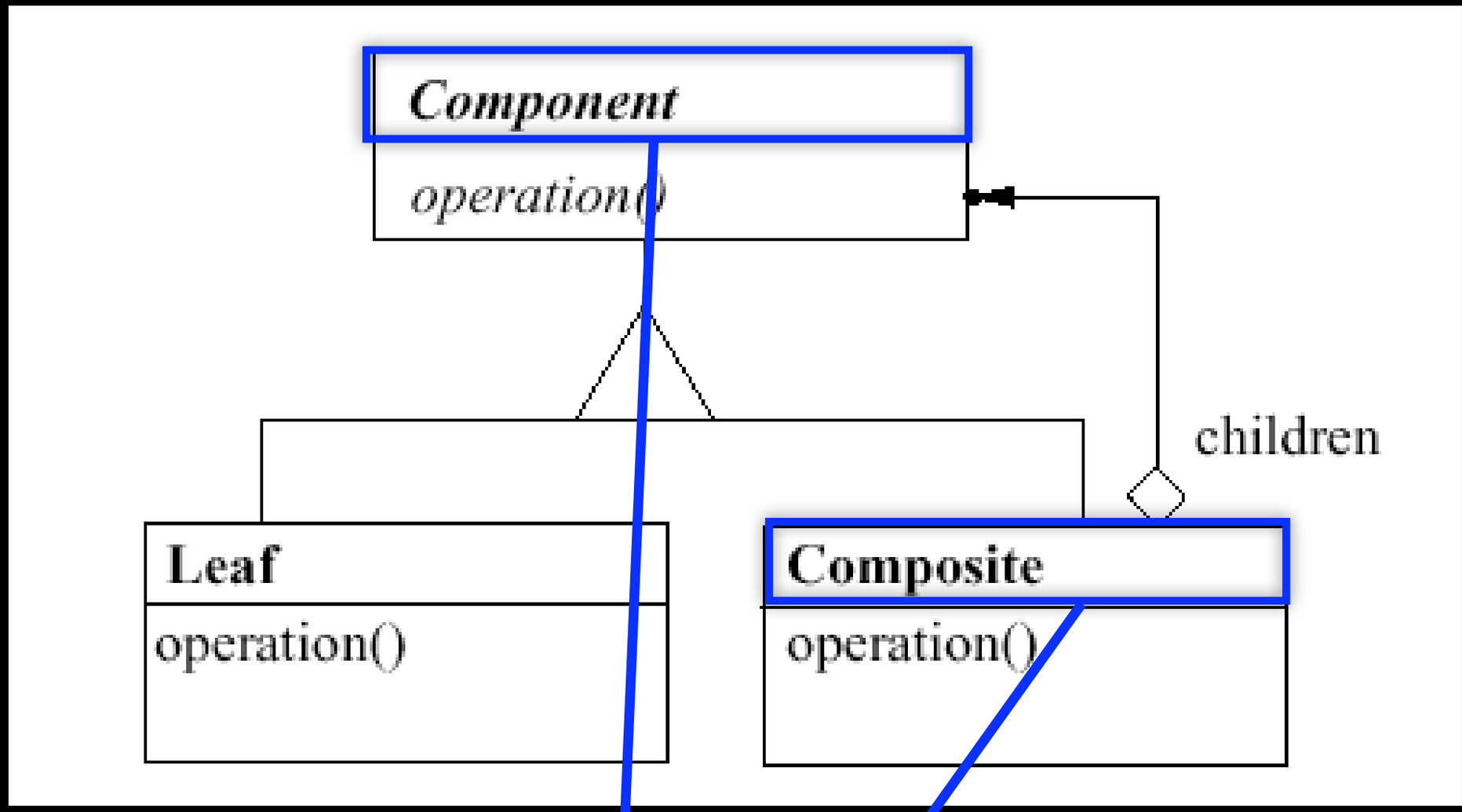


Example: composite design pattern



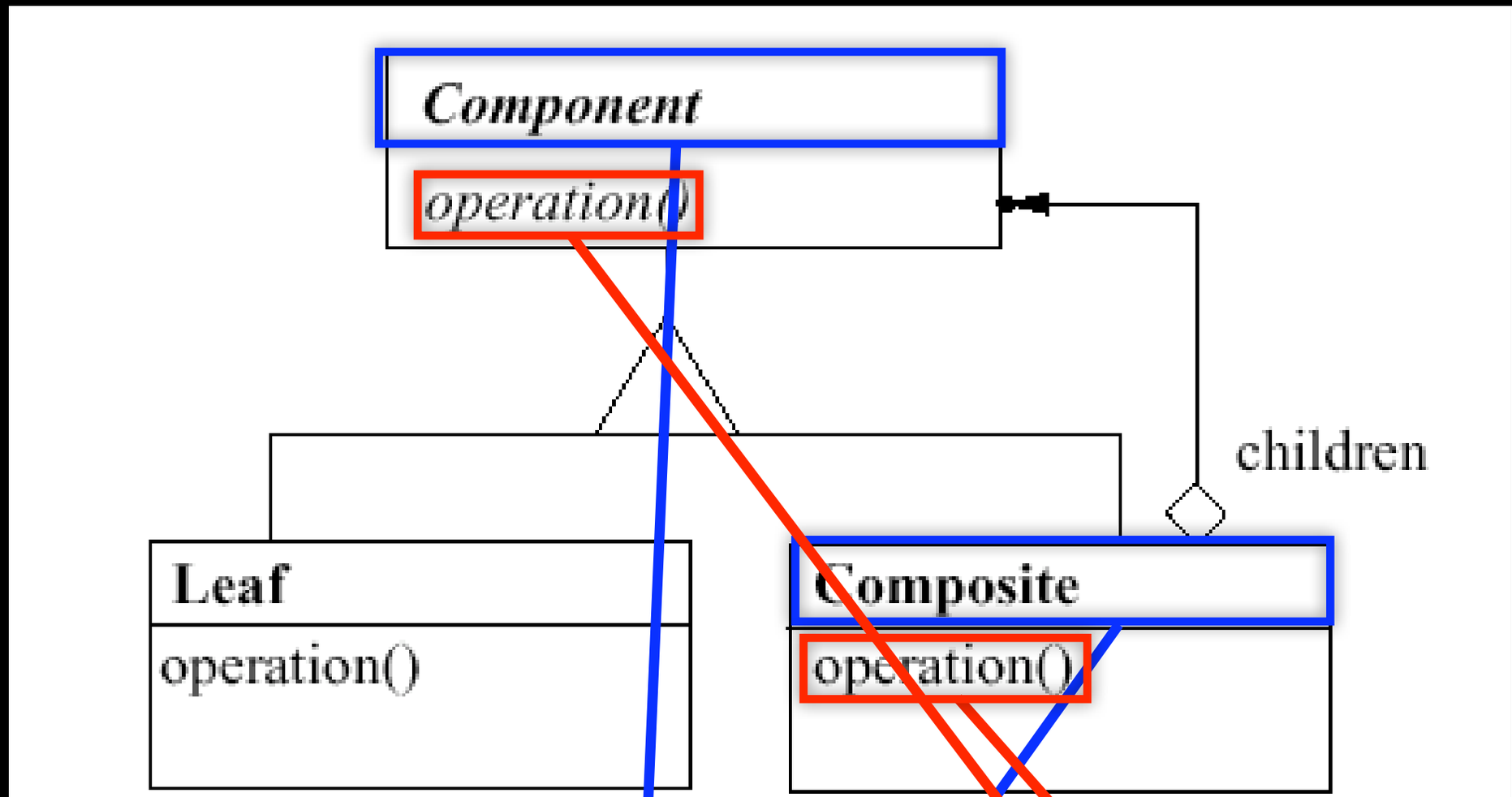

```
compositePattern(?comp, ?composite, ?msg) if
  compositeStructure(?comp, ?composite),
  compositeAggregation(?comp, ?composite, ?msg)
```

Example: composite design pattern



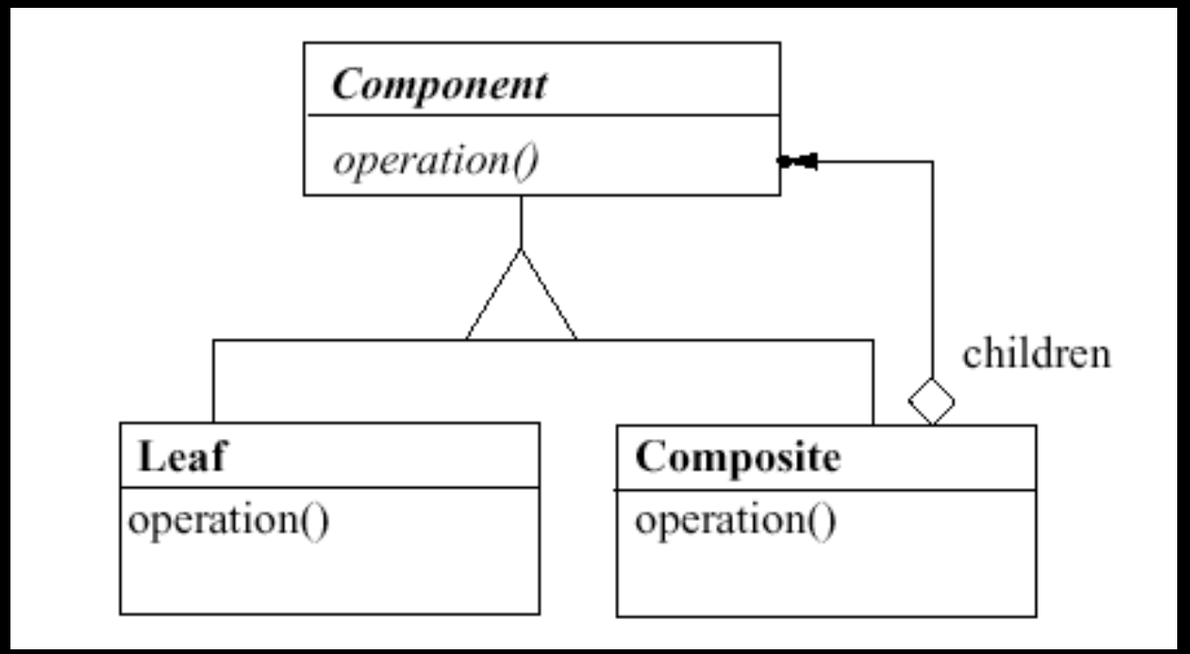

```
compositePattern(?comp, ?composite, ?msg) if
  compositeStructure(?comp, ?composite),
  compositeAggregation(?comp, ?composite, ?msg)
```

Example: composite design pattern




```
compositePattern(?comp, ?composite, ?msg) if
  compositeStructure(?comp, ?composite),
  compositeAggregation(?comp, ?composite, ?msg)
```

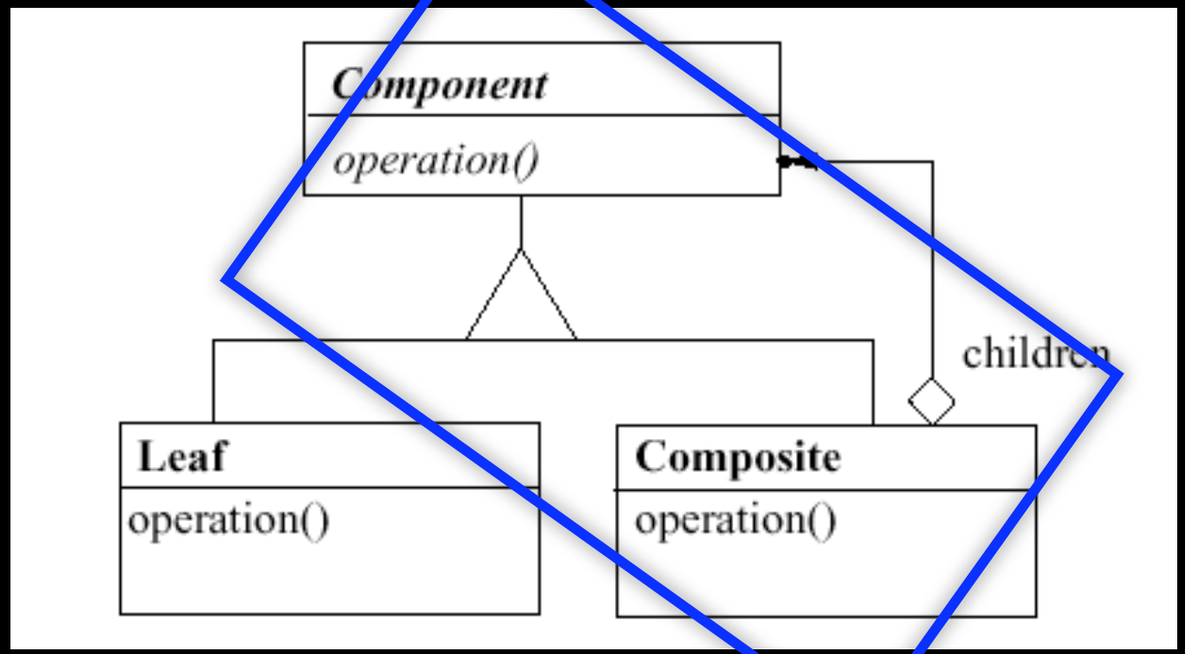
Example continued



Find structure

```
compositeStructure(?comp, ?composite) if
  class(?comp),
  classInHierarchyOf(?composite, ?comp)
```

Example continued



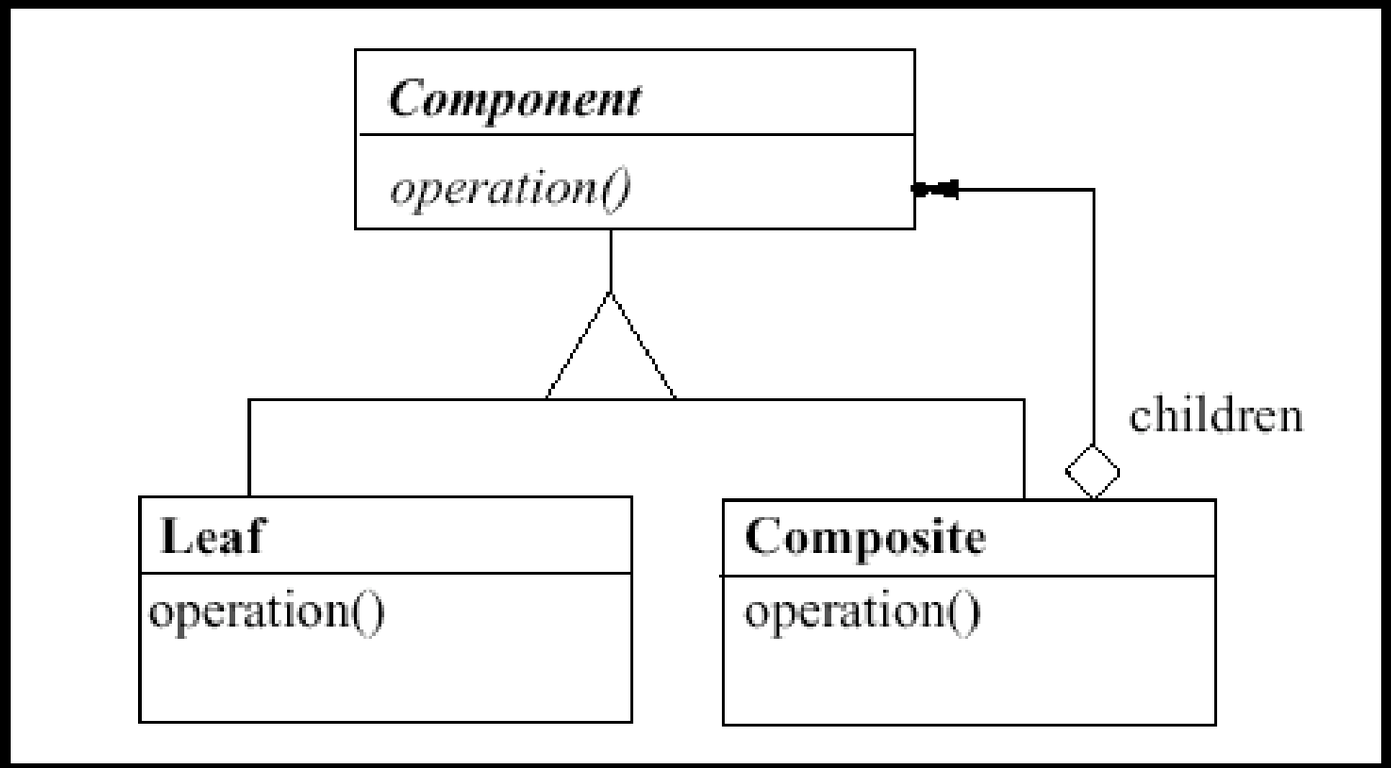
Find structure

```
compositeStructure(?comp, ?composite) if
  class(?comp),
  classInHierarchyOf(?composite, ?comp)
```

Example continued

```
Composite>>operation
   self children do:[:child |
         child operation]
```

Find aggregate

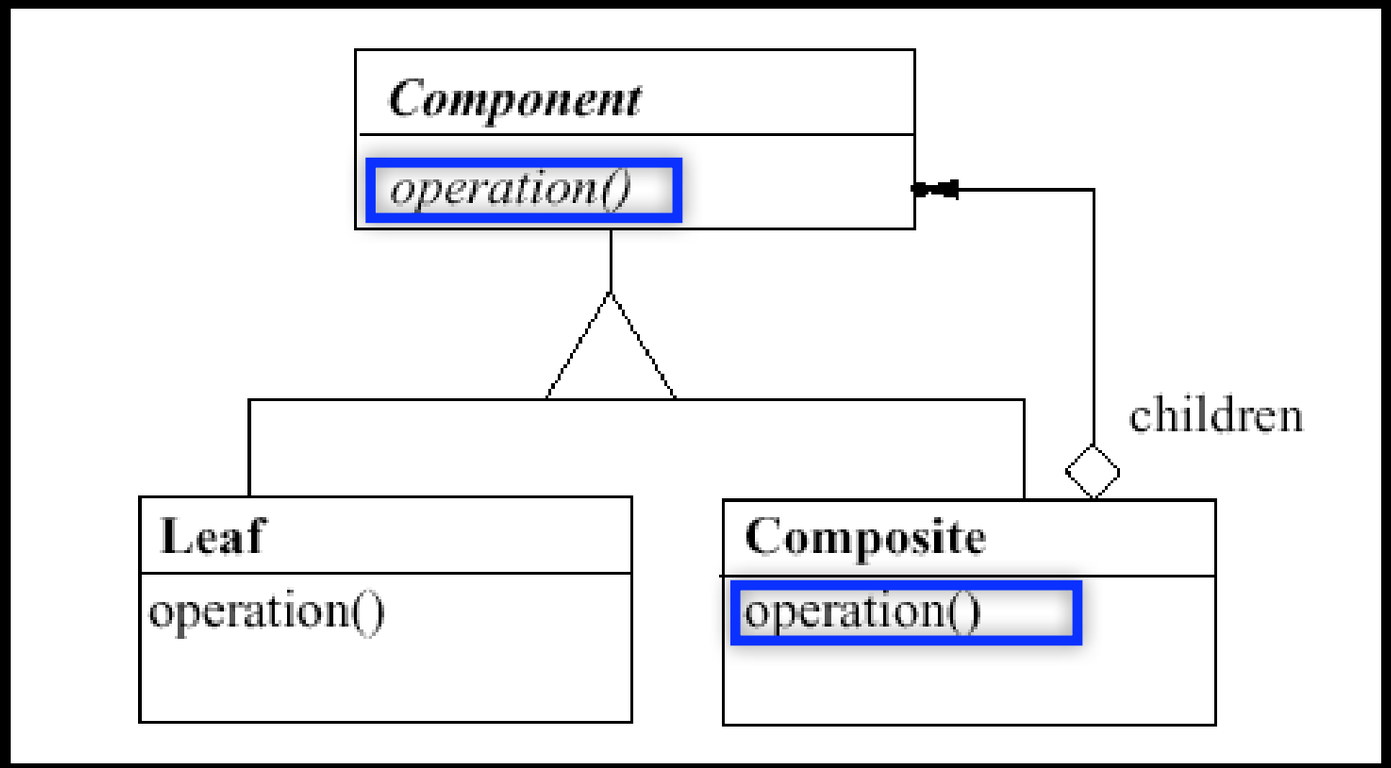



```
compositeAggregation(?comp, ?composite,?msg) if
   classesImplementCommonSelector(?comp,?composite),
   methodWithNameInClass(?method,?msg,?composite),
   statementsOfMethod(?statement,?method),
   iterationStatementWithBlock(?statement, ?iterationBlock),
   blockIteratesMessage(?iterationBlock, ?msg)
```

Example continued

```
Composite>>operation
   self children do:[:child |
         child operation]
```

Find aggregate

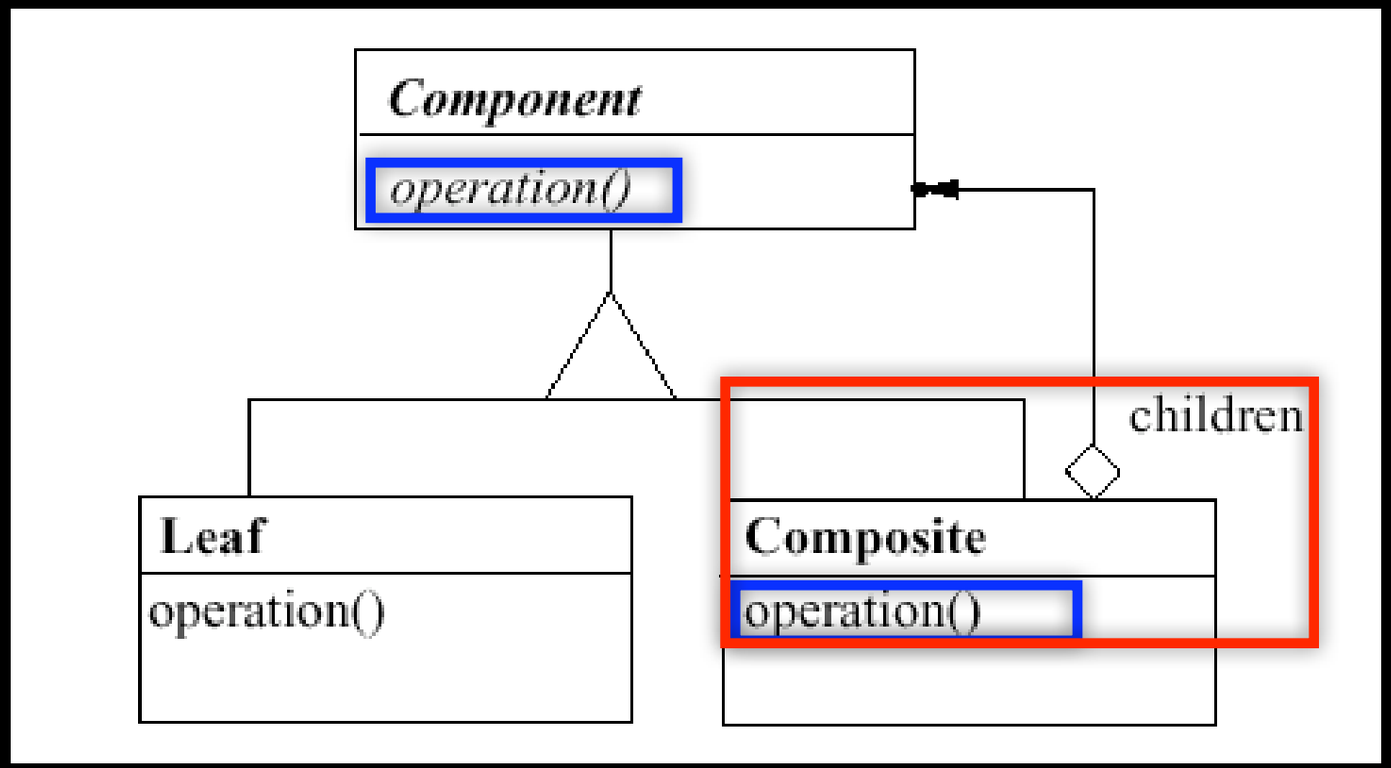



```
compositeAggregation(?comp, ?composite,?msg) if
  classesImplementCommonSelector(?comp,?composite),
  methodWithNameInClass(?method,?msg,?composite),
  statementsOfMethod(?statement,?method),
  iterationStatementWithBlock(?statement, ?iterationBlock),
  blockIteratesMessage(?iterationBlock, ?msg)
```

Example continued

```
Composite>>operation
   self children do:[:child |
         child operation]
```

Find aggregate




```
compositeAggregation(?comp, ?composite,?msg) if
  classesImplementCommonSelector(?comp,?composite),
  methodWithNameInClass(?method,?msg,?composite),
  statementsOfMethod(?statement,?method),
  iterationStatementWithBlock(?statement, ?iterationBlock),
  blockIteratesMessage(?iterationBlock, ?msg)
```

Composites in Smalltalk

```
[?comp-->[Refactory.Browser.LintRule],
?composite-->[Refactory.Browser.CompositeLintRule],
?msg-->[#resetResult]]
```

```
[?comp-->[VisualComponent],
?composite-->[CompositePart],
?msg-->[#updateSpot:]]
```

```
[?comp-->[Refactory.HotDraw.Figure],
?composite-->[Refactory.HotDraw.CompositeFigure],
?msg-->[#release]]
```

```
[?comp-->[XML.Node],
?composite-->[XML.Document],
?msg-->[#printOn:]]
```

```
[?comp-->[Refactory.Browser.NavigatorPart],
?composite-->[Refactory.Browser.TabNavigatorPart],
?msg-->[#navigator:]]
```

Queries through 7734 classes and 72962 methods in 13 minutes to find 81 solutions

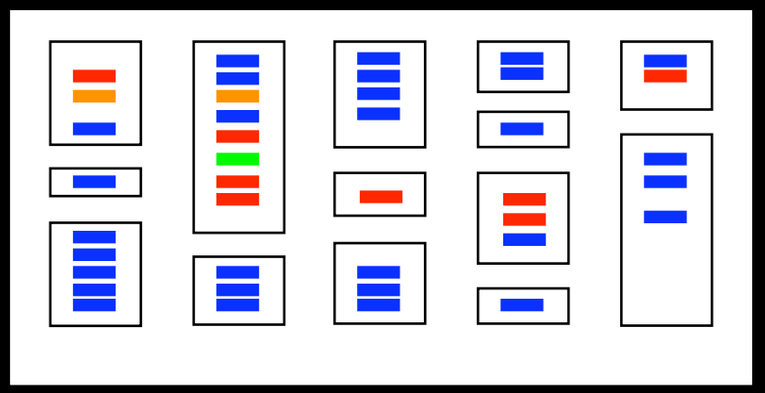
Applications

- **Applied to many problems:**
  - Co-evolution of design and source code
  - Reasoning about traces
  - Composing program generators
  - User interface specification
  - Aspect-oriented programming
  - Template querying

Applications

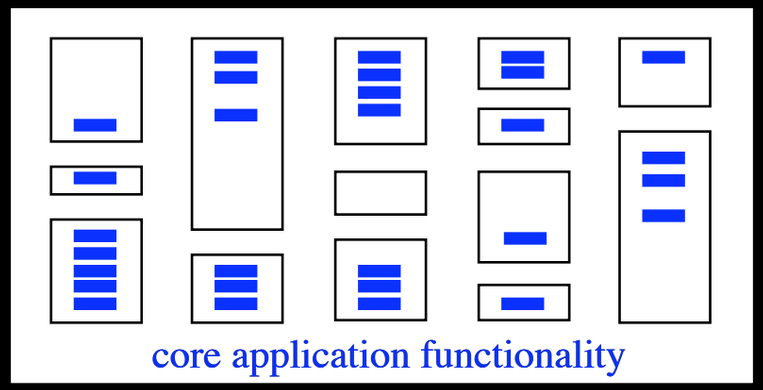
- **Applied to many problems:**
  - Co-evolution of design and source code
  - Reasoning about traces
  - Composing program generators
  - User interface specification
  - Aspect-oriented programming
  - Template querying

Aspect-oriented programming

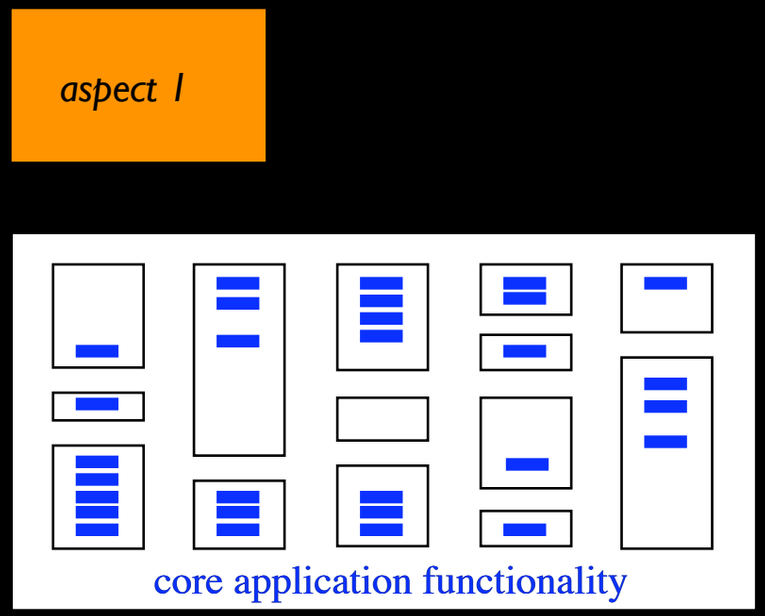


Aspect-oriented programming

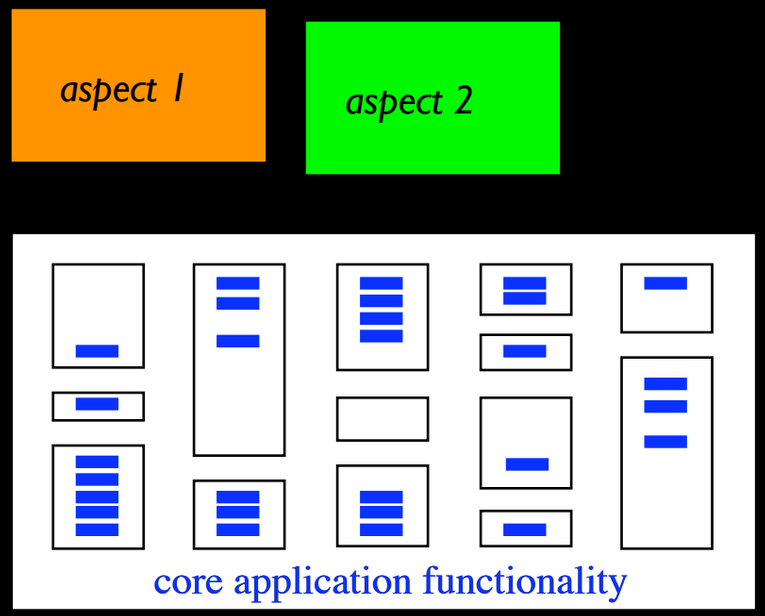
Aspect-oriented programming



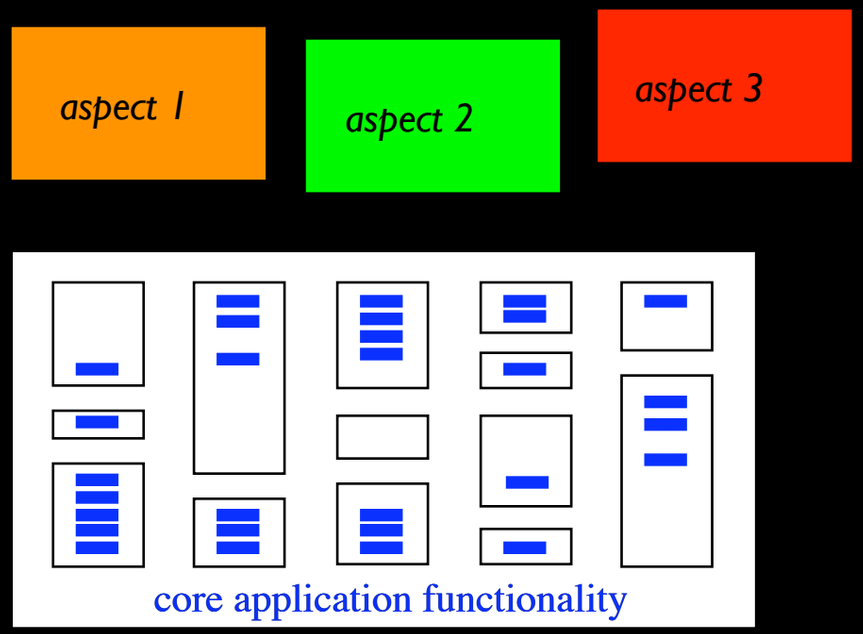
Aspect-oriented programming



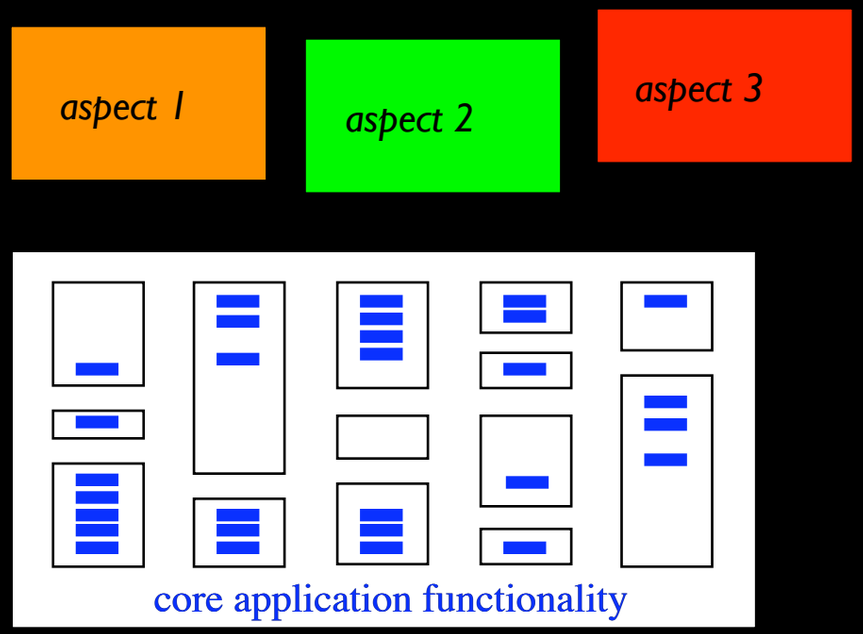
Aspect-oriented programming



Aspect-oriented programming

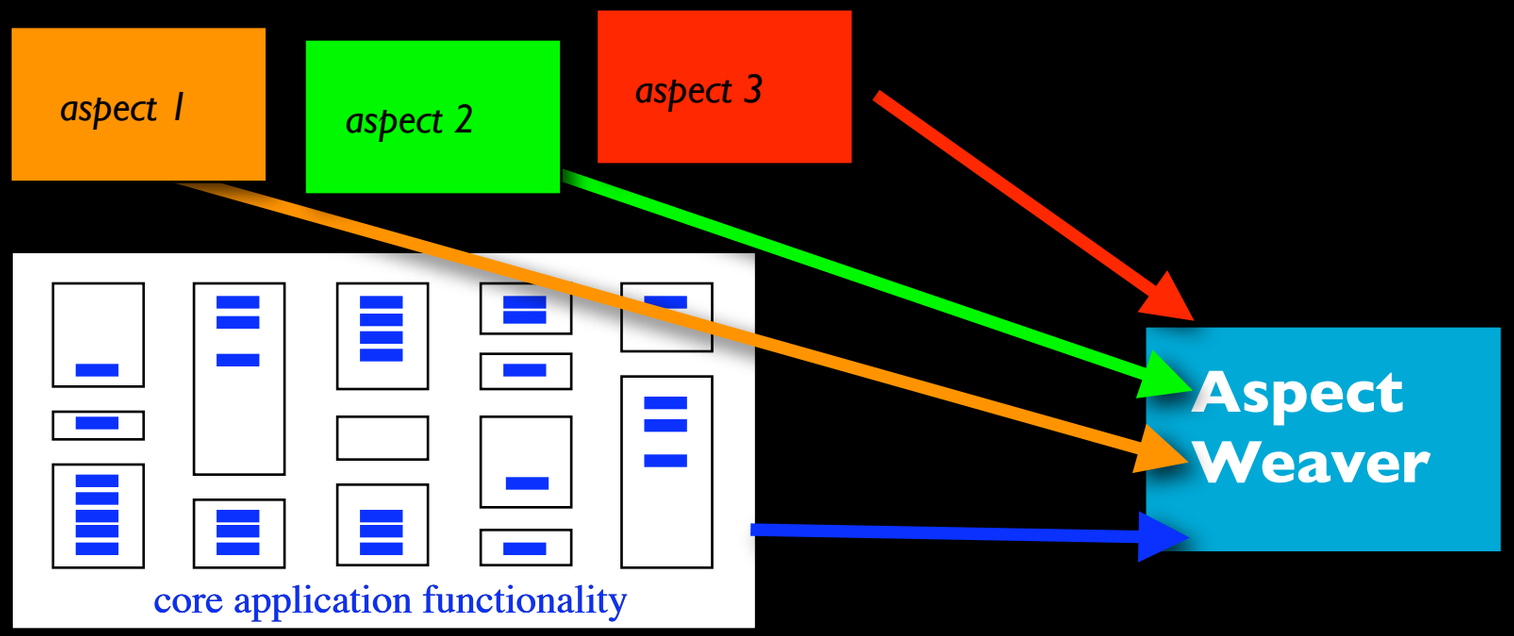


Aspect-oriented programming

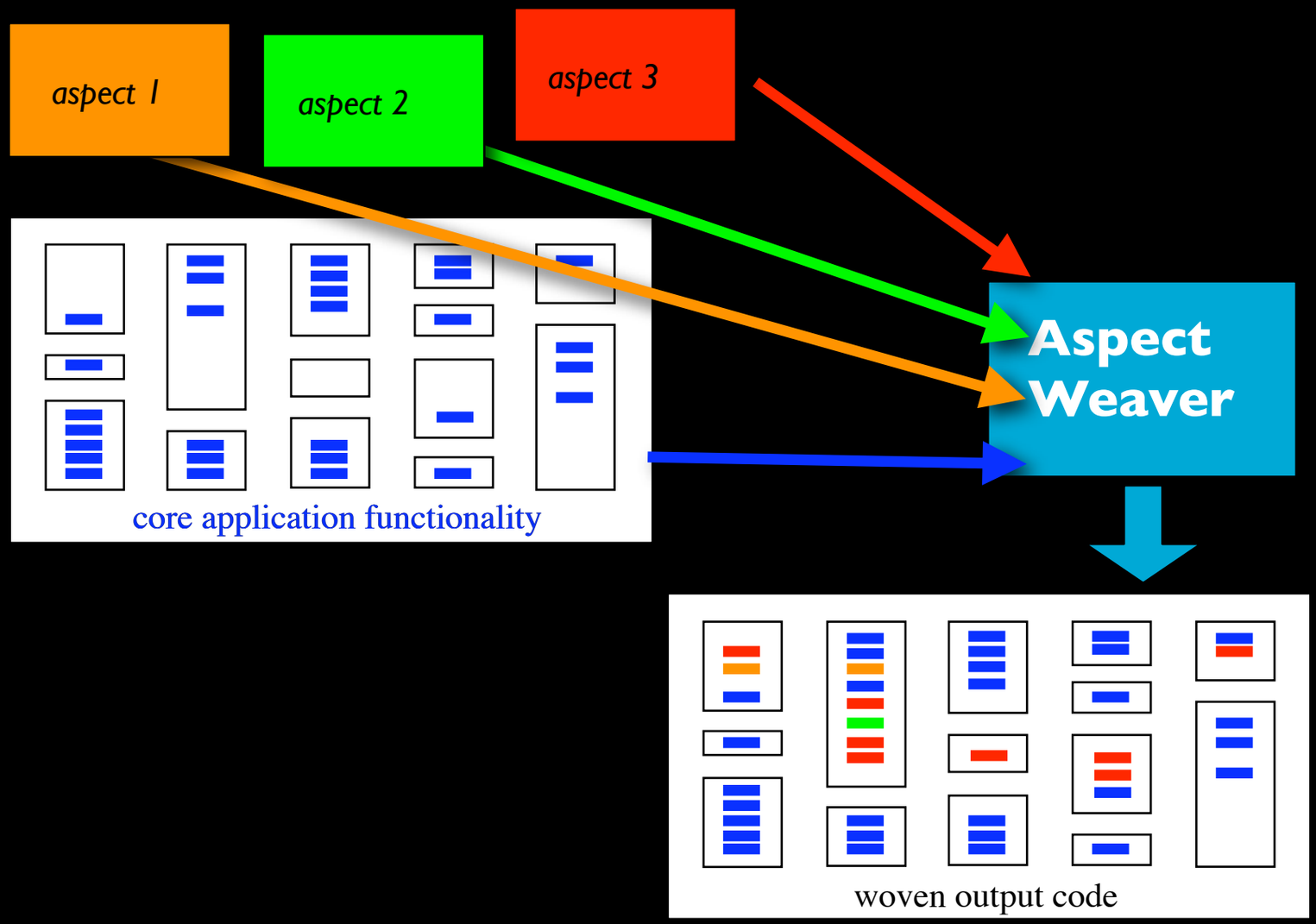


Aspect Weaver

Aspect-oriented programming



Aspect-oriented programming



CARMA [Gybels 03]

- **Logic meta programs as pointcuts**

Observer

```
changesState(?class, ?methodName) if
  shadowIn(?class, ?methodName, ?sp),
  assignmentShadow(?sp, ?variable)

changesState(?class, ?methodName) if
   shadowIn(?class, ?methodName, ?sp),
   messageShadow(?sp, ?rcvr, ?msg),
   selfReceiver(?rcvr),
   changesState(?class, ?msg)

after ?jp matching
   reception(?jp, ?msg, ?args),
   inObject(?jp, ?obj),
   objectClass(?obj, ?class),
   changesState(?class, ?msg),
   not(caller(?jp, ?obj))
do
   observers notify
```

CARMA [Gybels 03]

- **Logic meta programs as pointcuts**

Observer

```
changesState(?class, ?methodName) if
  shadowIn(?class, ?methodName, ?sp),
  assignmentShadow(?sp, ?variable)

changesState(?class, ?methodName) if
   shadowIn(?class, ?methodName, ?sp),
   messageShadow(?sp, ?rcvr, ?msg),
   selfReceiver(?rcvr),
   changesState(?class, ?msg)
```

```
after ?jp matching
   reception(?jp, ?msg, ?args),
   inObject(?jp, ?obj),
   objectClass(?obj, ?class),
   changesState(?class, ?msg),
   not(caller(?jp, ?obj))
do
   observers notify
```

All methods that change the state of an object

CARMA [Gybels 03]

- **Logic meta programs as pointcuts**

Observer

```
changesState(?class, ?methodName) if
  shadowIn(?class, ?methodName, ?sp),
  assignmentShadow(?sp, ?variable)

changesState(?class, ?methodName) if
   shadowIn(?class, ?methodName, ?sp),
   messageShadow(?sp, ?rcvr, ?msg),
   selfReceiver(?rcvr),
   changesState(?class, ?msg)
```

All methods that change the state of an object

```
after ?jp matching
   reception(?jp, ?msg, ?args),
   inObject(?jp, ?obj),
   objectClass(?obj, ?class),
   changesState(?class, ?msg),
   not(caller(?jp, ?obj))
do
   observers notify
```

Pointcut: intervene at which run-time points?

CARMA [Gybels 03]

- **Logic meta programs as pointcuts**

Observer

```
changesState(?class, ?methodName) if
  shadowIn(?class, ?methodName, ?sp),
  assignmentShadow(?sp, ?variable)

changesState(?class, ?methodName) if
   shadowIn(?class, ?methodName, ?sp),
   messageShadow(?sp, ?rcvr, ?msg),
   selfReceiver(?rcvr),
   changesState(?class, ?msg)
```

All methods that change the state of an object

```
after ?jp matching
   reception(?jp, ?msg, ?args),
   inObject(?jp, ?obj),
   objectClass(?obj, ?class),
   changesState(?class, ?msg),
   not(caller(?jp, ?obj))
do
   observers notify
```

Pointcut: intervene at which run-time points?

Advice: what to do?

Template querying

- Write down a piece of code with variables in it as a query
- Use static analysis to increase precision

```
if jtClassDeclaration(?c){
class ?c {
   private ?type ?field;
   public ?type ?name() { return ?field; }}
}
if jtClassDeclaration(?c){
   class ?c {
   private ?type ?field;
   public ?rt ?name(?type ?var) { ?field = ?var; }}
}
```

Declarative Meta Programming

- **For more information:**
  - http://prog.vub.ac.be/DMP
  - http://prog.vub.ac.be/SOUL