Agenda

- Introduction
- Motivation
- Problems and their solution
- Conclusion
Introduction

- Based on bachelor thesis of Jan Pikl
  - Documentation:  http://jpikl-xweaver.blogspot.com
  - Sources:        http://sourceforge.net/projects/clexw/

- Extending XWeaver
  - Cross-Language Extension for XWeaver

- Aspects

- XWeaver and AspectX
  - Source level weaving
Separation of Concerns

- **Functional part of the application (say, control)**
- **Non-functional part of the application (say, failure detection and failure recovery)**

Model B superimposed upon Model A

- **Non-functional part cross-cuts functional part**
Source weaving

Base Code with no Error Handling

```c
... fred_1();
... fred_2();
... fred_3();
...```

Base Code with Error Handling Policy #1 i.e. logging

```c
... if (fred_1() != 0 )
    printf(“Error found“\n);
... if (fred_2() != 0 )
    printf(“Error found“\n);
... if (fred_3() != 0 )
    printf(“Error found“\n);
...```

Base Code with Error Handling Policy #2 i.e. application reset

```c
... if (fred_1() != 0 )
    doSoftwareReset();
... if (fred_2() != 0 )
    doSoftwareReset();
... if (fred_3() != 0 )
    doSoftwareReset();
...```
Based on bachelor thesis of Jan Pikl

Extending XWeaver Aspects

XWeaver Architecture

XML Model of Base Code (in target language)

AspectX Program (for target language)

Modified Code (in target language)

Weaving Rules

Compiler

Weaver

Locator

Annotated Base Code

XWeaver Compiler Weaver.xmlexmlexmlexmlexmlexmlxml

XML Model of Base Code (in target language)
XWeaver Meta Weaving Approach

- **OO Code Meta-Model**
  - C/C++ Code
  - srcML Model
  - Meta-level weaving

- **Meta-Level Weaver**
  - C/C++ Code
  - IBM C/C++ Weaver
  - Instantiation

- **Java Code XML Model**
  - Java Code
  - XML Model
  - Java weaving

- **Java Weaver**
  - Java weaving
  - Instantiation
Motivation

- **Main idea for GAČR**
  - Identify features is specification such that are cross cutting concerns, then model them and then even later implement them via AOP
  - Cross-cutting through different languages at model level, designer doesn't want to be restricted by a particular implementing (base) language thus particular by aspect language too
  - Little attention in published papers on AOP

- **Model Driven Development**
- **Reusable aspects code**
- **Aspects based on UML**
Challenges

- **Commonalities** between languages?
  - Java, C++, C#, Ada

- **Differences** between languages?
  - Java, C++, C#, Ada

- How to describe aspects?
- Create new aspect oriented language?
- Use XML as AspectX?
Solved problems

- Extracted commonalities
- Designed new aspect language
  - ANTLR
- Memory management
  - 4 different approaches to memory deallocation in C/C++ vs. GC in Java
Cross-Language AspectX

- CLAX → XSLT → AspectX
  - CLAX as frontend for XWeaver
- Language format
  - aspect
  - pointcut
  - advice
  - include
- Code Modifier Language
Aspect

<?xml version="1.0" encoding="UTF-8" ?>
<aspect xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://clexw.sf.net/CrossLanguageAspectX
    ../..xsd/CrossLanguageAspectX.xsd"
    xmlns="http://clexw.sf.net/CrossLanguageAspectX"
    name="Sample Aspect">

    <description>
        <p>This is sample aspect</p>
    </description>

    <pointcut type="method" name="setterMethods"> ... </pointcut>
    <pointcut type="method" name="printMethods"> ... </pointcut>
    <pointcut type="method" name="targetMethods"> ... </pointcut>
    <advice type="begin" name="printMethodName"> ... </advice>
</aspect>
Simple Pointcut

```xml
<pointcut type="method" name="setterMethods">
  <constraint type="name" value="set.*" />
</pointcut>
```
<pointcut type="method" name="printMethods">
  <and>
    <constraint type="name" value="print" />
    <not>
      <restriction type="within">
        <pointcut type="class">
          <constraint type="name" value="OutStream" />
        </pointcut>
      </restriction>
    </not>
  </and>
</pointcut>
<pointcut type="method" name="targetMethods">
  <or>
    <pointcutRef ref="setterMethods" />
    <pointcutRef ref="printMethods" />
  </or>
</pointcut>
<advice type="begin" name="printMethodName">
  <pointcutRef ref="targetMethods" />
  <codeModifier type="codeFragment">
    println "Entering ${functionName}";
  </codeModifier>
</advice>
Java

```
Factory fac = Factory.getInstance();
Item it = fac.createItem("abc");
if (it == null) {
    throw NullPointerException();
} else {
    it.doSomething(123);
}
```

C++

```
Factory * fac = Factory::getInstance();
Item * it = fac->createItem("abc");
if (it == NULL) {
    throw new NullPointerException();
} else {
    it->doSomething(123);
}
```
Cross-Language AspectX

- CLAX → XSLT → AspectX
- CLAX as frontend for XWeaver
- Language format
  - aspect
  - pointcut
  - advice
  - include

Code Modifier Language

XML Model of Base Code
(in target language)

XSLT

Locator

Annotated Base Code

XML Model of Target Language

Compiler

AspectX Program
(for target language)

Weaving Rules

Modified Code
(in target language)

Weaver

Annotated Base Code

XML Model of Target Language

XWeaver Architecture
Open Issues

- Modeling tool is missing
- Implemented only for C++ and Java
- C# is pending
- Ada probably never
  - No support from srcML
  - Proof of concept verified for Ada
Conclusion

- All problems at code level were addressed
- Including test scenarios
- “The best bachelor thesis ever”
  - [Ondřej Rohlík]

- No paper, yet
Questions?

Thank You