Softlang Working Group
http://softlang.wikidot.com/

MegaL Traceabiltiy Recovery

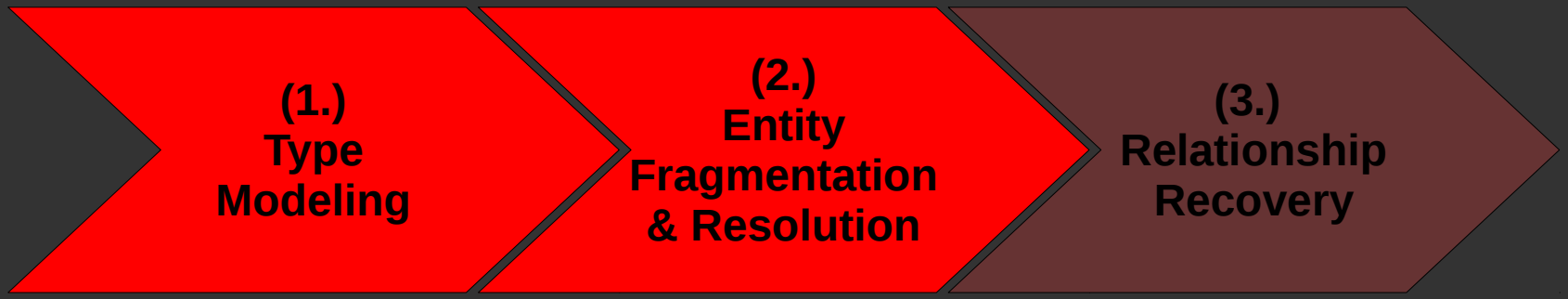
https://github.com/maxmeffert/megal-tr

Fragment Inference & *ANTLR* Backed Plugin API

Meeting 2016-06-28

University of Koblenz-Landau
Maximilian Meffert

Quick Recap



(1.) Models the fragment ***types*** of the domain

(2.) Extracts the fragments from a domain instance (*Fragmentation*) & resolves further specializations if necessary (*Resolution*)

(3.) Recovers relationships between fragments
[not discussed now]

University of Koblenz-Landau
Maximilian Meffert

Quick Recap

```
JavaFragment < Fragment

// type declarations
JavaClass < JavaFragment
JavaInterface < JavaFragment
JavaEnum < JavaFragment

// member declarations
JavaInnerClass < JavaFragment
JavaMethod < JavaFragment
JavaConstructor < JavaMethod
JavaField < JavaFragment
JavaAnnotation < JavaFragment
```

```
public class Foo {

    static public class Bar {

        private void getBar () {

        }

    }

    private String bar;

    public String getBar() {
        return bar;
    }

    public void setBar(String bar) {
        this.bar = bar;
    }

}
```

Quick Recap

```
aJavaFile: File
aJavaFile elementOf Java
aJavaFile = 'workspace:/org.softlang.megal.plugins/input/Foo.java'

aJavaFile.Foo#0: JavaClass
aJavaFile.Foo#0 partOf aJavaFile
aJavaFile.Foo#0 = 'file:/.../Foo.java#/0/Foo/JavaClass'

aJavaFile.Foo#0.Bar#0: JavaInnerClass
aJavaFile.Foo#0.Bar#0 partOf aJavaFile.Foo#0
aJavaFile.Foo#0.Bar#0 = 'file:/.../Foo.java#/0/Foo/JavaClass/0/Bar/JavaInnerClass'

aJavaFile.Foo#0.Bar#0.getBar#0: JavaMethod
aJavaFile.Foo#0.Bar#0.getBar#0 partOf aJavaFile.Foo#0.Bar#0
aJavaFile.Foo#0.Bar#0.getBar#0 = 'file:/.../Foo.java#/0/Foo/JavaClass/0/Bar/JavaInnerClass/0/getBar/JavaMethod'

aJavaFile.Foo#0.bar#1: JavaField
aJavaFile.Foo#0.bar#1 partOf aJavaFile.Foo#0
aJavaFile.Foo#0.bar#1 = 'file:/.../Foo.java#/0/Foo/JavaClass/1/bar/JavaField'

aJavaFile.Foo#0.getBar#2: JavaMethod
aJavaFile.Foo#0.getBar#2 partOf aJavaFile.Foo#0
aJavaFile.Foo#0.getBar#2 = 'file:/.../Foo.java#/0/Foo/JavaClass/2/getBar/JavaMethod'

aJavaFile.Foo#0.setBar#3: JavaMethod
aJavaFile.Foo#0.setBar#3 partOf aJavaFile.Foo#0
aJavaFile.Foo#0.setBar#3 = 'file:/.../Foo.java#/0/Foo/JavaClass/3/setBar/JavaMethod'
```

Fragmentation Result

Fragment Model & KB

```
public class Foo {

    static public class Bar {

        private void getBar () {

        }

    }

    private String bar;

    public String getBar() {
        return bar;
    }

    public void setBar(String bar) {
        this.bar = bar;
    }

}
```

A computational fragment model should be loosely based on syntax trees.
Scope defines parthood.

Fragment Model & KB

- Fragments build a simple generic tree alongside the original AST
  - A leaf node is called ***atom***
  - A non-leaf node is called ***compound***
- A Fragment KB (*Fragments*) exists separately from the Megamodel KB during an evaluation process

Fragment Model & KB

Qualified Fragment Names

```
aJavaFile.Bar#0.[…].doStuff#666
```

Name of the declared entity
Short name of the fragment
Index of the fragment in its container

Fragment Model & KB

- ***QFNs*** are used as identifies for the derived entities
- ***QFNs*** depict *partOf* relationships
- ***Indexes*** depict the position of fragments in their respective composite

Fragment Model & KB

Generic URI Form:
`scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]`

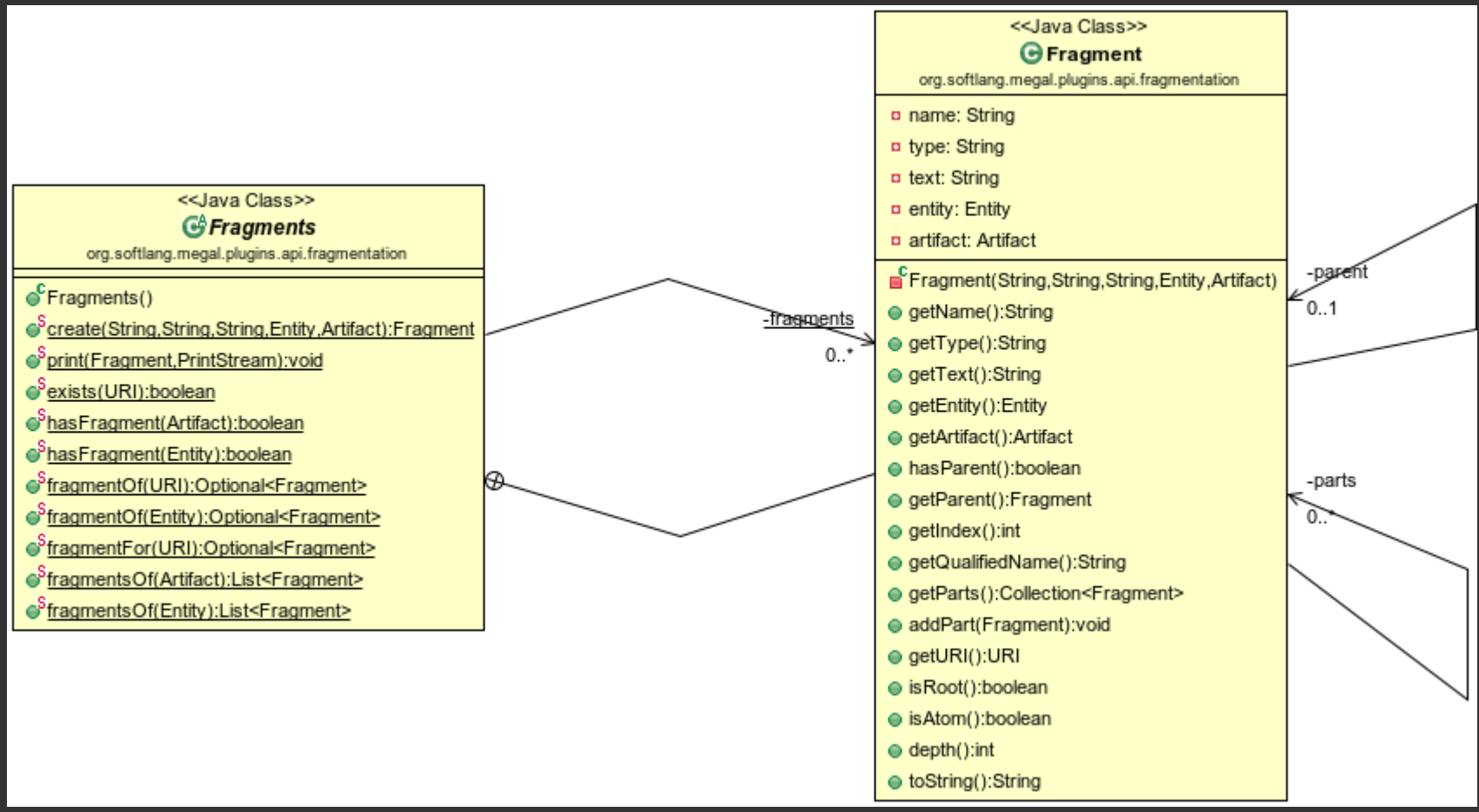
Fragment URI Form:
scheme://location#fragment

Where **fragment** conforms to:

```
Fragment  : '/' INDEX '/' NAME '/' TYPE ('/' Fragment )*
INDEX     : \d+
NAME      : \w+
TYPE      : \w+
```

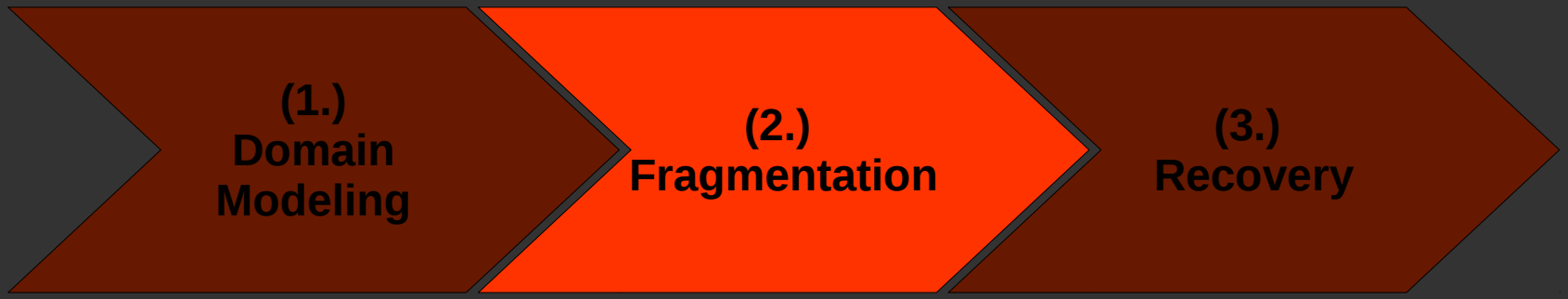
file://path/to/Foo.java#/0/Foo/JavaClass/2/getBar/JavaMethod

Fragment Model & KB



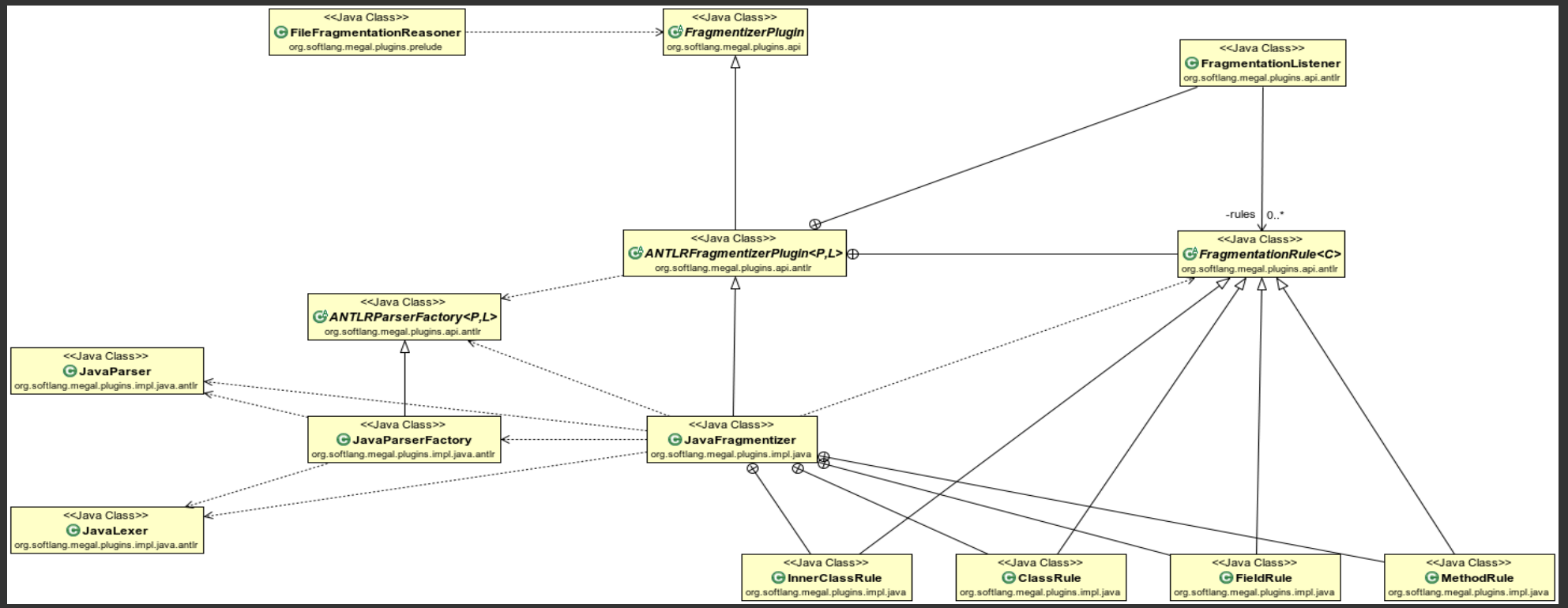
3. ANTLR Backed Plugins

ANTLR Based Fragmentation



- asdf

ANTLR Based Fragmentation API



XML-Dialect Resolution

XML Fragmentation

Motivation

A domain model is a collection of (axiomatic) *statements* over entity-types and entities.

Motivation


```
Foo < Entity
Bar < Entity

partOf < Foo * Bar

...
```

Motivation

Can we provide singular plugins for recovery?

TODO