

MegaL

Traceability Link Recovery

Declared Traceability Links

```
model Test  
import Prelude
```

```
Foo : Artifact  
Bar : Artifact
```

```
Foo correspondsTo Bar
```

```
Foo = "/path/to/foo1.ext"  
Foo = "/path/to/foo2.ext"  
Bar = "/path/to/bar1.ext"  
Bar = "/path/to/bar2.ext"
```

Binding 4 Artifacts
=
Declaring 4 Links

Test.megal

Declared Traceability Links

Input: *Megamodel* m

Output: *TraceabilityLinks* tls

```
Foreach (Relationship rel in m) {  
    Links lLinks := allLinks(rel.left)  
    Links rLinks := allLinks(rel.right)  
    Foreach (Link lLink in lLinks) {  
        Foreach (Link rLink in rLinks) {  
            tls.add(TraceabilityLink(rel, lLink, rLink))  
        }  
    }  
}
```

Pseudo code recovery algorithm for **explicit** Traceability Links

Declared Traceability Links

```
correspondsTo("/path/to/foo1.ext", "/path/to/bar1.ext")  
correspondsTo("/path/to/foo1.ext", "/path/to/bar2.ext")  
correspondsTo("/path/to/foo2.ext", "/path/to/bar1.ext")  
correspondsTo("/path/to/foo2.ext", "/path/to/bar2.ext")
```

Recovered Traceability Links

Inferred Traceability Links

The idea is to recover/infer Traceability Links such as:

- `correspondsTo("/path/to/fool.ext", "/path/to/bar1.ext")`
 - `correspondsTo("/path/to/fool.ext/fooPartN", "/path/to/bar1.ext/barPartI")`
 - ...
 - `correspondsTo("/path/to/fool.ext/fooPartM", "/path/to/bar1.ext/barPartJ")`

Inferred Traceability Links

Additional Info is needed to infer traceability links

```
model Prelude
...
File < Artifact
Folder < Artifact
Fragment < Artifact
...
partOf < Artifact * Artifact
partOf < Artifact * Technology
partOf < Technology * Technology
partOf < Language * Technology
...
elementOf < File * Language
```

Prelude.megal

Inferred Traceability Links

Step 1

```
model Test
  FooBarTech : Technology

  FooLang : Language
  FooLang partOf FooBarTech

  BarLang : Language
  BarLang partOf FooBarTech
```

Test.megal

Declaring “top-level” composite entities for a technology model

Inferred Traceability Links

Step 2

```
Foo : Artifact
Foo elementOf FooLang
Bar : Artifact
Bar elementOf BarLang

Foo correspondsTo Bar

Foo = "/path/to/foo1.ext"
Foo = "/path/to/foo2.ext"
Bar = "/path/to/bar1.ext"
Bar = "/path/to/bar2.ext"
```

Test.megal

Declaring **explicit** traceability Links

Declaring artifacts as components of a language to make fragments retrievable

Inferred Traceability Links

Step 3

```
FooPart : Fragment
```

```
FooPart partOf Foo //would imply FooPart elementOf FooLang?
```

```
BarPart : Fragment
```

```
BarPart partOf Bar //would imply BarPart elementOf BarLang?
```

```
FooPart correspondsTo BarPart //needed?
```

Test.megal

Declaring **implicit** traceability Links
Declaring fragment artifacts as components of other Artifacts

Inferred Traceability Links

Do we base traceability link inference
on partOf/elementOf hierarchies?