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

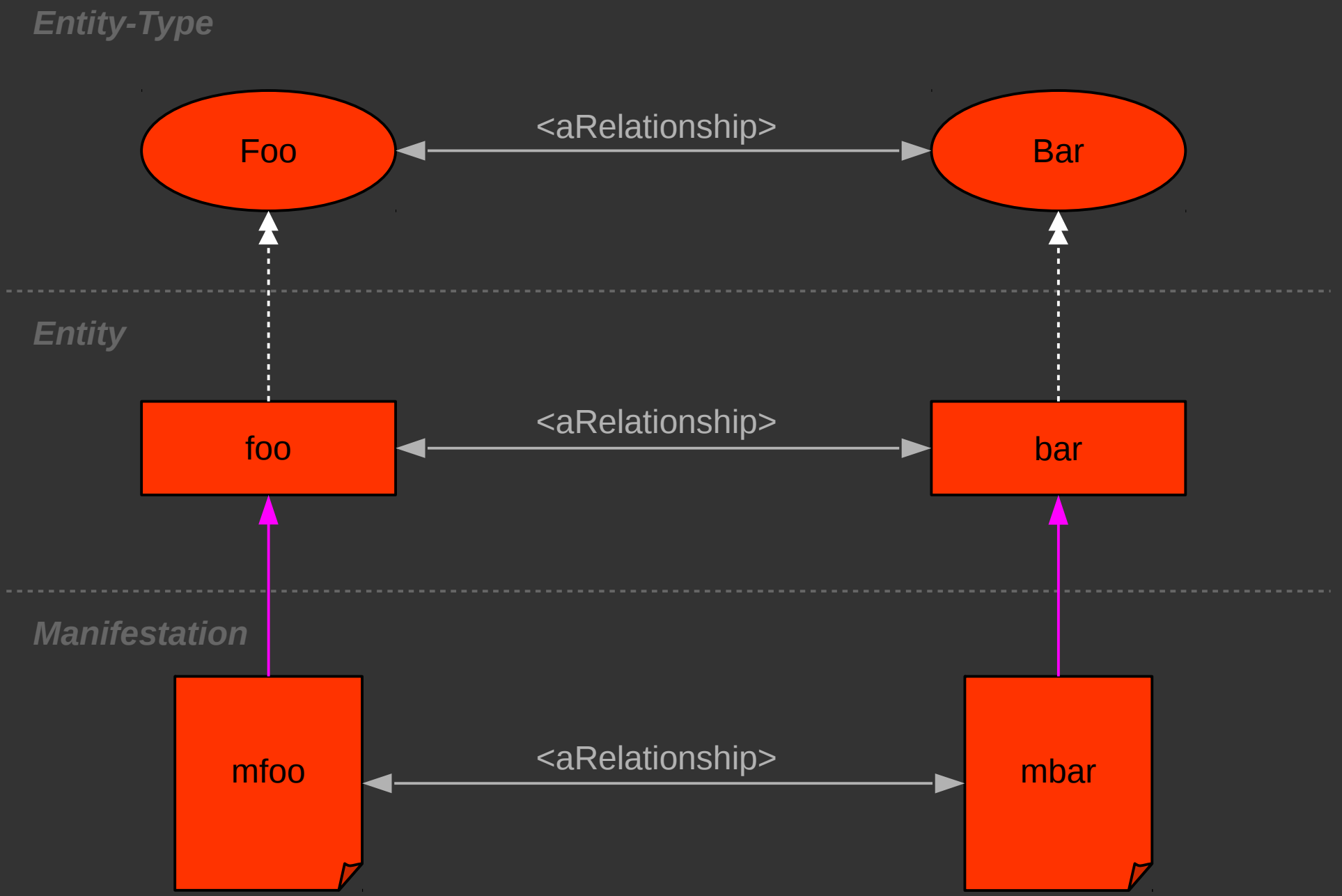
MegaL Traceabiltiy Recovery

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

Formal Foundations

Meeting 2016-03-15

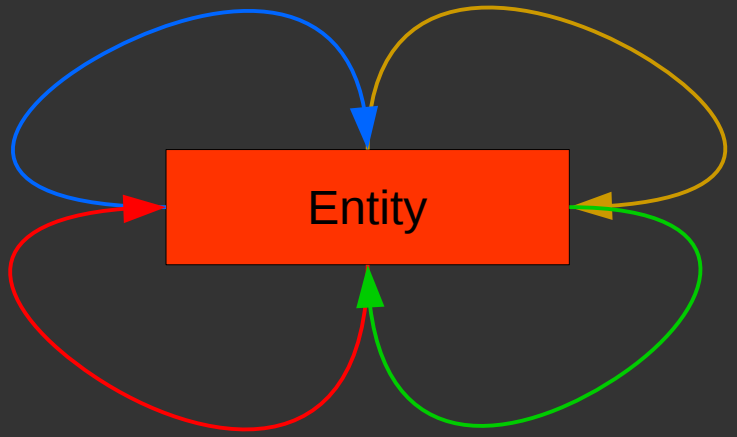
University of Koblenz-Landau
Maximilian Meffert


Entity-Type
Foo
<aRelationship>
Bar
Entity
foo
<aRelationship>
bar
Manifestation
mfoo
<aRelationship>
mbar

Prelude

partOf

isA



correspondsTo

conformsTo

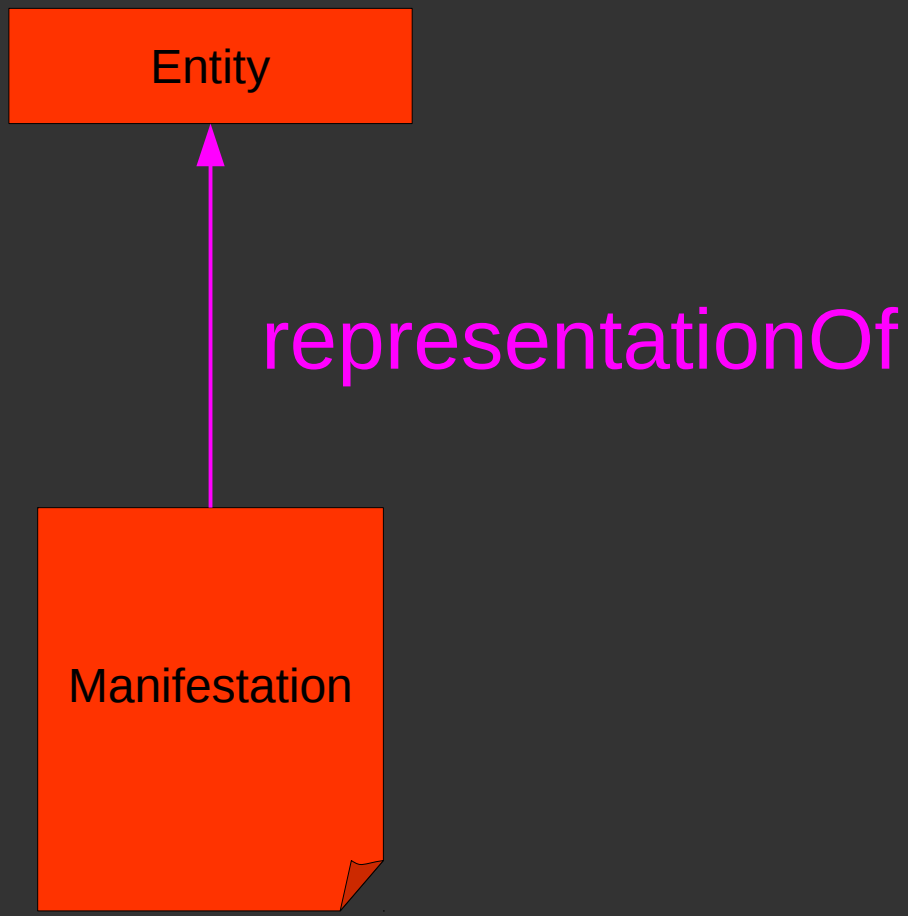
Prelude

„Signum ∈ significat est. Ita a ∈ b legitur a est quoddam b“

„The Symbol ∈ means is. Thus a ∈ b is read ***a is a b***“

(Giuseppe Peano: Arithmetices principia nova methodo exposita, 1889, S. X)

Prelude



Motivation

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

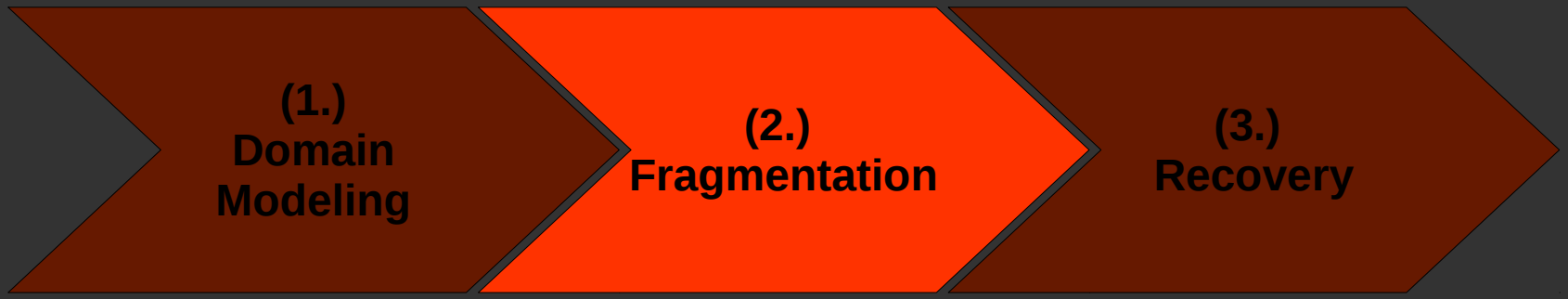
Motivation


```
Foo < Entity
Bar < Entity

partOf < Foo * Bar

...
```

Motivation



Extends a KB with entities found in Manifestations.

Fragmentation is done by several plugins, specifically tailored to a domain.

Motivation

Can we provide singular plugins for recovery?

Unambiguous Definitions?

A partOf B $:\leftrightarrow$???

A conformsTo B $:\leftrightarrow$???

A correspondsTo B $:\leftrightarrow$???

A representationOf B $:\leftrightarrow$???

First Order Logic

Logic Symbols

Quantifier symbol:	$\forall, \exists$
Logic Operator symbols:	$\neg, \wedge, \vee, \leftrightarrow, \rightarrow$
Description Operator symbol:	ι
Variable symbols:	$a, b, c, \ldots, x, y, z$ (lower case)
Equality/Identity symbol:	$=$
Parentheses symbols:	$( )$ $[ ]$

Non-logic Symbols

Predicate symbols:	$A, B, C, \ldots, X, Y, Z$ (upper case)

Terms

Any variable is a term.

Formulars

asdf

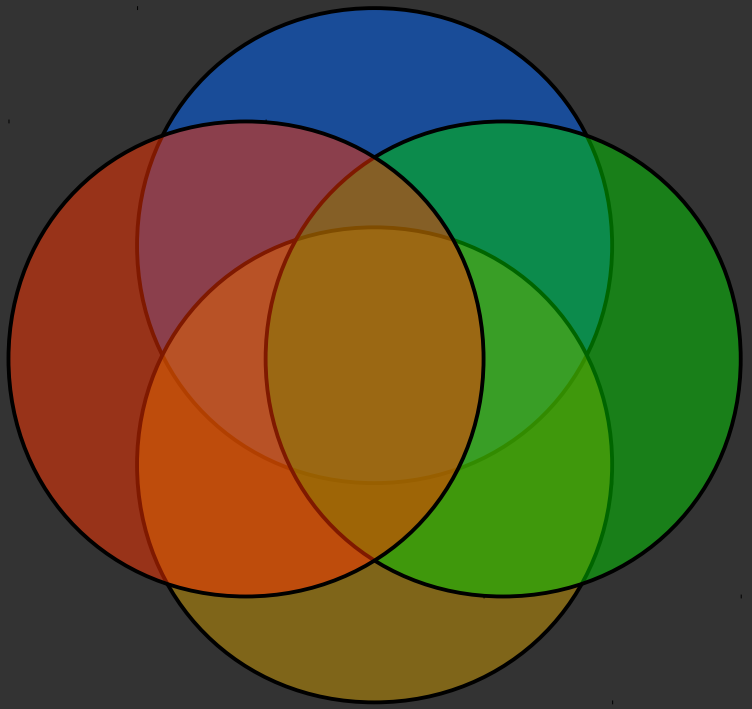
Mereology

The study of
wholes and their constituent ***parts***

Mereology

Excursus

Logic



Computer Science

Philosophy

Mathematics

Mereology

An axiomatic system with focus
on the ***parthood*** (partOf) predicate:

P x y

“x is part of y”

Ground Mereology

(Reflexive)

$$P\ x\ x$$

(Antisymmetric)

$$P\ x\ y \wedge P\ y\ x \rightarrow x = y$$

(Transitive)

$$P\ x\ y \wedge P\ y\ z \rightarrow P\ x\ z$$

Mereotopology

Excursus

An axiomatic extension of a Mereology with predicates Φ other than parthood

$$\Phi\ x\ y$$

And statements like

$$P\ x\ y \rightarrow \Phi\ x\ y$$
$$\Phi\ x\ y \rightarrow P\ x\ y$$
$$P\ x\ y \leftrightarrow \Phi\ x\ y$$
$$\ldots$$

University of Koblenz-Landau
Maximilian Meffert

Formalization #1

Formalization #1 : Example

`Foo < Entity`	$Foo \subset Entity$
`Bar < Entity`	$Bar \subset Entity$
`partOf < Foo * Bar`	$\forall (x,y) \in Foo \times Bar (P\ x\ y)$
`foo : Foo`	$foo \in Foo$
`bar : Bar`	$bar \in Bar$
`foo partOf bar`	$P\ foo\ bar$
`foo = mfoo`	$R\ mfoo\ foo$
`bar = mbar`	$R\ mbar\ bar$

Trace Recovery Rule

$$\Phi\ x\ y \wedge R\ a\ x \wedge R\ b\ y \rightarrow \Phi\ a\ b$$

Recovers the predicate Φ between two manifestations a and b.

(Assume: $\Phi \neq R$)

Formalization #1

Problem:

*Types are **not** handled properly!*

Typed Predicate Logic

References

(1) asdf