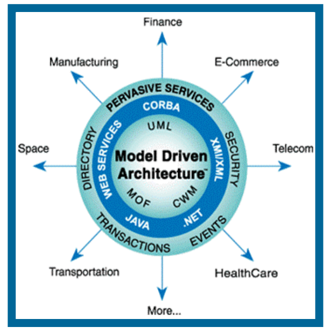
The MOF Representation of the IFF



Introduction

The Information Flow Framework (IFF) is a very large framework for defining semantics. There are thousands of terms and axioms. Included will be the semantics of ontologies, the semantics of programs, etc. Although very large, the IFF architecture has a principled foundation and has a very regular construction. There is a sharp boundary between the object-level and the metalevel. The object level is intended for content, whereas the metalevel is intended for structure – all structure. The original aim of the metalevel construction was to represent the intuitions of the working category-theorist. In particular, a central goal was to be able to completely axiomatize the notion of a standard category, such as the category of small sets, the category of topologies and continuous maps, the category of groups and group homomorphisms, etc.

As such, the IFF metalevel is divided into three metalevels corresponding to the set-theoretic distinction between the small, the large and the generic (sets, set-theoretic classes and collections). The lower metalevel uses large (set-theoretic class-level) notions (such as set-theoretic classes, large relations, standard categories, large classifications, etc.) and defines set-level notions (such as first order logic (FOL) languages, FOL theories, FOL model-theoretic models, FOL logics, etc.). The upper metalevel uses the very few generic (collection-level) notions (such as collections, generic functions, generic relations, etc.) and defines large (set-theoretic class-level) notions (such as set-theoretic classes, large relations, standard categories, large classifications, etc.).

In addition, at least up until now, the IFF uses no additional data types beyond sets. This includes natural numbers, although we recognize that eventually these will also be needed for more detailed studies. To recapitulate, (1) the IFF is axiomatized, (2) the IFF is very regular, and (3) the IFF uses nothing beyond sets, function and binary relations. As such, it seems to be ideally set up for application of the Meta Object Facility (MOF) of the OMG. Such an application should aid immensely in the implementation of the IFF. In order to get started, we start small with our aim to provide MOF metamodels for the notions of IFF languages and IFF theories.

The MOF Representation of IFF Languages

The notion of an IFF first order logic (FOL) language is discussed in the language namespace (‘**`lang`**’) of the IFF Ontology (meta) Ontology. IFF languages only deal with type information. They and their morphisms form a category called Language. The notion of an IFF language is like an aligned notion of hypergraph. *Entity* types are synonymous with (hypergraph) nodes; *relation* types are synonymous with (hyper) edges. Moreover, following tradition, the names in hypergraphs are called (logical) *variables*, and there is an explicit referencing (sorting) function from variables (names) to entity types (nodes).

First, we give a concise mathematical definition (Figure 1), second we discuss and formalize the various parts of this definition, and third we construct a MOF metamodel for languages. An IFF *language* $\boldsymbol{L} = \langle \mathit{refer}(\boldsymbol{L}), \mathit{sign}(\boldsymbol{L})\rangle$ (Figure 1) consists of

- a *reference* function $*_L = \mathit{refer}(\boldsymbol{L})$
- a *signature* function $\partial_L = \mathit{sign}(\boldsymbol{L})$

that satisfy the pullback constraint that “the target of the signature of a language is the signature set of the reference function of the language”

$\mathit{tgt}(\mathit{sign}(\boldsymbol{L})) = \mathit{sign}(\mathit{refer}(\boldsymbol{L}))$.

$$\begin{tikzcd}
\mathit{rel}(\boldsymbol{L}) \arrow[r, "\#_L"] \arrow[dr, "\partial_L"'] & \wp\mathit{var}(\boldsymbol{L}) & \mathit{var}(\boldsymbol{L}) \arrow[d, "*_L"'] \\
 & \mathit{sign}(*_L) \arrow[u, "\mathit{refer\text{-}arity}(\boldsymbol{L})"] & \mathit{ent}(\boldsymbol{L})
\end{tikzcd}$$

Figure 1: Type Language

The code below has two kinds of terminology – internal and external. An internal term is a term axiomatized in the ambient namespace. An external term is axiomatized in some other namespace, possibly in some other meta-ontology. Internal terms are displayed as is – they do not have a namespace prefix adornment. External terms are displayed in the form

`namespace-prefix$external-term`

where ‘`external-term`’ is the name of the term as axiomatized in the namespace referenced by the preferred namespace prefix ‘`namespace-prefix`’. In particular, in the code immediately below the external term ‘`SET$class`’ refers to the notion of a set-theoretic class as axiomatized in the class namespace in the IFF Upper Core (meta) Ontology. Whereas, the external term ‘`set.ftn$function`’ refers to the notion of a function between sets as axiomatized in the function namespace in the IFF Lower Core (meta) Ontology.

The reference and signature terms represent functions in two different senses. In the code below an IFF language is a parameter, and reference and signature are set-theoretic class functions mapping that language to small notions. In both cases, that small notion is a function between small sets. There is a composition operator for large functions, and this is used to represent the pullback constraint mentioned above.

```
(1) (SET$class language)

(2) (SET.FTN$function reference)
    (= (SET.FTN$source reference) language)
    (= (SET.FTN$target reference) set.ftn$function)

(3) (SET.FTN$function signature)
    (= (SET.FTN$source signature) language)
    (= (SET.FTN$target signature) set.ftn$function)

(4) (= (SET.FTN$composition [reference set.ftn$signature])
       (SET.FTN$composition [signature set.ftn$target]))
```

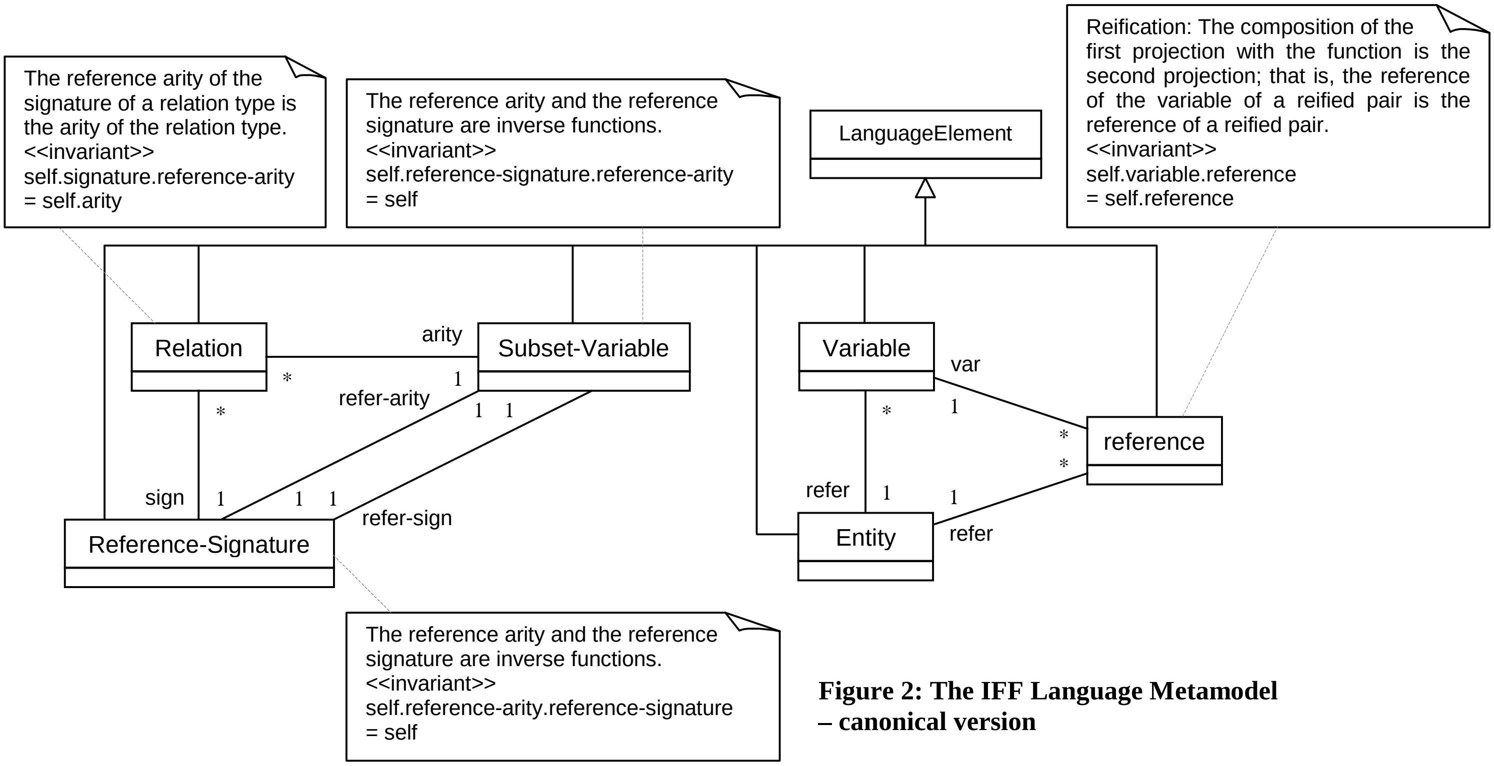


Figure 2: The IFF Language Metamodel – canonical version

Figure 2 is a MOF metamodel diagram that represents the notion of an IFF language. MOF metamodel diagrams are in the IFF lower metalevel, and they only deal with small notions. The language parameter corresponds to the MOF metamodel parameter. The metamodel in Figure 2 is a straightforward translation of Figure 1. Any set (parameterized by a language) is represented by a meta-class and any function is represented by a meta-relation. The main semantic constraint (upper left comment) states that the composition of the signature and the reference-arity functions is the arity function. Since the reference arity function is a bijection, its inverse has been added in order to be able to specify the inverse semantics (middle two comments). The reification of the reference function has been added in order to provide a parameter for the reference signature set construction that is axiomatized in the IFF Lower Core (meta) Ontology. The standard semantic constraint for reified functions is added (upper right comment).

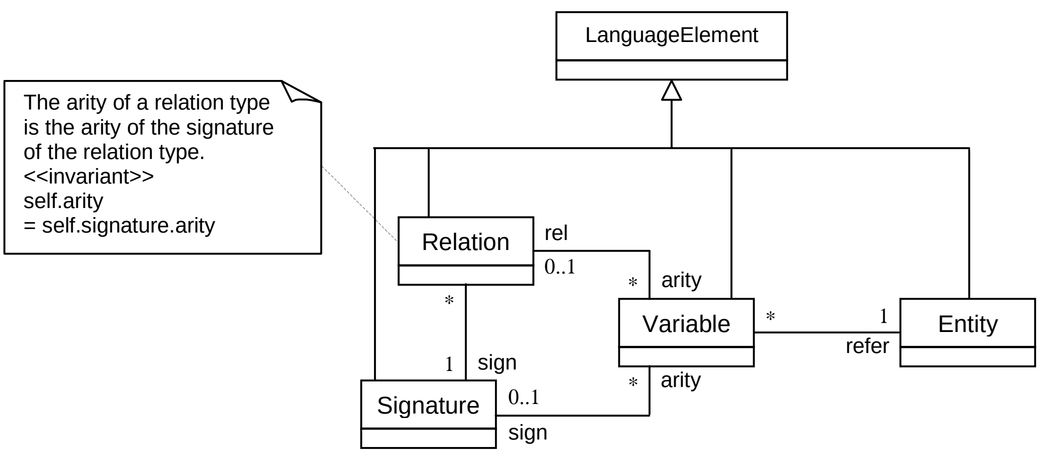


Figure 3: The IFF Language Metamodel – streamlined version

Figure 3 demonstrates that there are many ways to represent an IFF notion as an MOF metamodel. This is clearly a streamlined version of an IFF language metamodel. Here the notion of an IFF arity has been absorbed into a MOF relation-variable meta-relation. There is also a similar (but more general) MOF signature-variable meta-relation. The signature set here is the reference signature set, but the reference function was not reified. Hence, a parameter for the signature set construction has not been provided.

The MOF Representation of IFF Language Morphisms

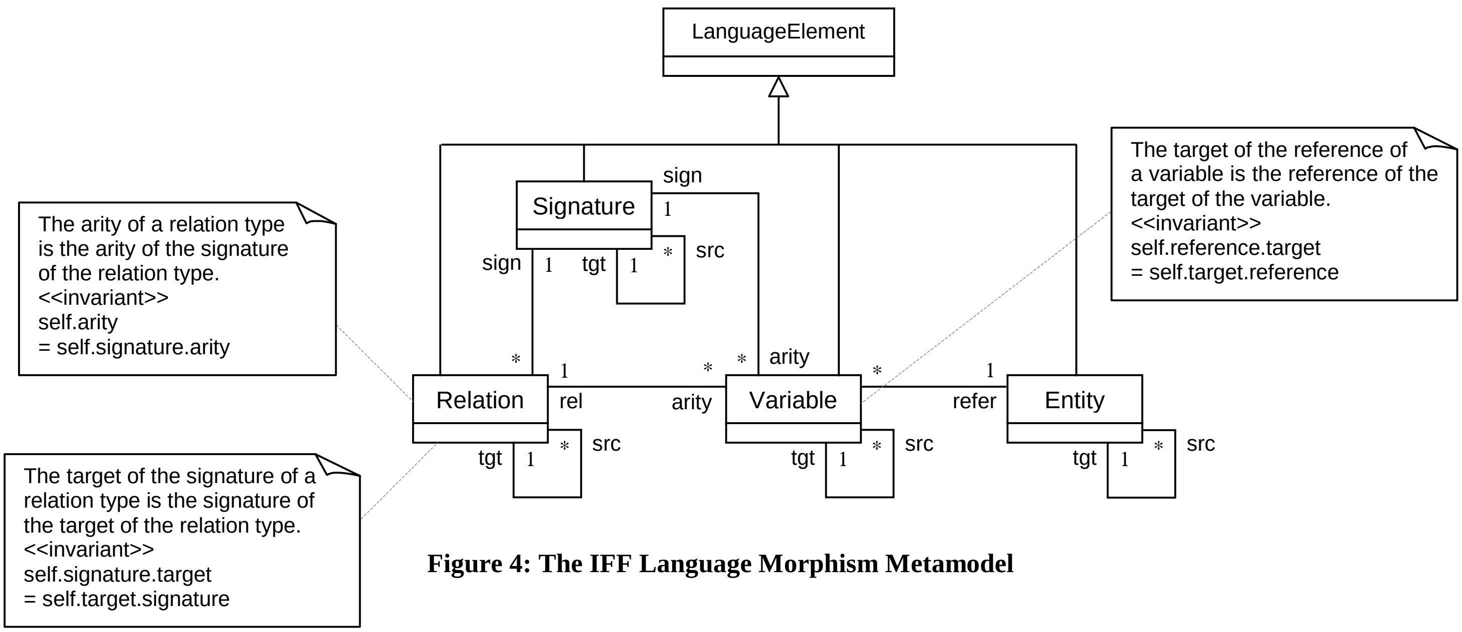


Figure 4: The IFF Language Morphism Metamodel