OKBC: A Programmatic Foundation for Knowledge Base Interoperability

Background: This paper was written when author at SRI International, together with researchers in Stanford University. This paper discuss technical design issues faced in the development of OKBC, highlight how OKBC improves upon its predecessor GFP, and report on practical experiences in using it.

Knowledge bases (KBs) can be constructed by the assembly of prefabricated knowledge components. Knowledge components include both pieces of domain knowledge (for example, theories of economics or fault diagnosis) and KB tools (for example, editors and theorem provers).

Most of the current KB development tools can only manipulate knowledge residing in the **knowledge representation system (KRS)** for which the tools were originally developed. **Open Knowledge Base Connectivity (OKBC)** is an application programming interface for accessing KRSs, and was developed to enable the construction of reusable KB tools. OKBC improves upon its predecessor, the Generic Frame Protocol (GFP), in several significant ways.

- OKBC can be used with a much larger range of systems because its knowledge model supports an assertional view of a KRS.
- OKBC provides an explicit treatment of entities that are not frames, and it has a much better way of controlling inference and specifying default values.
- OKBC can be used on practically any platform because it sup ports network transparency and has implementations for multiple programming languages.

OKBC is an application programming interface (API) for KRSs that has been developed to address the problem of KB tools **reusability**. A KRS can be bound to OKBC by defining a mapping from OKBC to the native API of that KRS. To achieve interoperability, a KB tool accesses a KRS using only OKBC operations. Such a tool is isolated from the peculiarities of the KRS and can be used with any KRS that has been bound to OKBC.

The OKBC specification consists of three components:

- a knowledge model
- a collection of operations to access a KRS
- a collection of behaviors

The design experience of OKBC conclude as below two points:

A. An expressiveness vs. generality tradeoff: The expressiveness of the knowledge model must be controlled so that it can work with a range of KRSs.

    -If the knowledge model is too expressive, it becomes difficult to define OKBC bindings for systems that have limited representational power.

    -If the knowledge model is insufficiently expressive, the OKBC bindings for systems with more representational power will not expose their capabilities.

B. Second, the protocol was augmented by two features to support variability among KRSs: additional returned values and behaviors.

Wherever it is not feasible to legislate certain requirements, KRSs can expose the difference either globally by setting the value of a behavior or locally by returning an additional value from

an operation.

The OKBC Knowledge Model

The OKBC includes includes constants, frames, slots, facets, classes, individuals, and knowledge bases.

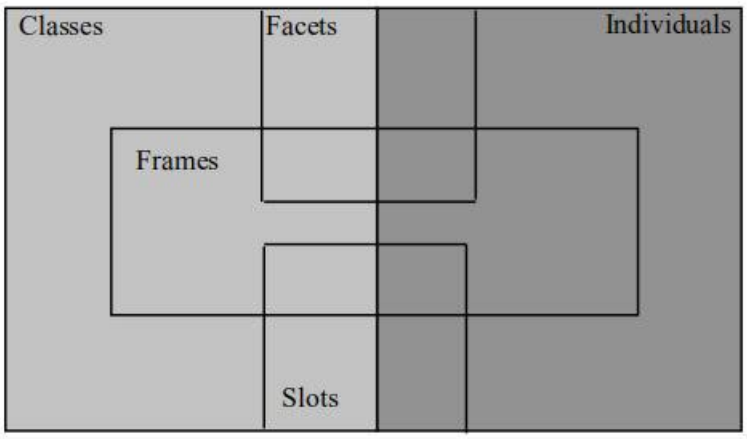

Figure 1: The OKBC knowledge model defines that classes and individuals form disjoint partitions of a KB. It does not commit to whether classes, individuals, slots, and facets are represented as frames. It also does not commit to whether slots and facets should be represented as classes or individuals.

The model expand the range of supported KRSs:

a. Support for assertions: GFP was found to be inadequate for use with KRSs that prefer to view a KB as a collection of logical sen tences, as well as systems that have a knowledge model more expressive than the knowledge model of GFP. To address these problems, we introduced a tell/ask inter face that supports an assertional view of a KB.
b. Assertion Language OKBC defines an assertion language (AL) for declarative specification of knowledge. The AL is a first-order language with conjunction and predicate symbols.
c. Assertions not guaranteed to be supported by OKBC are also considered and OKBC defines operations to handle assertions outside of the AL
d. Handling entities that are not frames problems below are also considered and solved
    a) Not all entities are frames
    b) Not all entity categories are frames

The model design and practical anaysis also includes below considerations:

- Controlling KRS inference
- Handling defaults
- Expanding the range of applications

Experiences in using OKBC

This part argue two main points:

A. OKBC has been successful in its goal of enabling the construction of inter operable tools by

presenting empirical evidence based on the definition of OKBC bindings for several KRSs.

a) OKBC bindings
b) Binding a less expressive KRS
c) Binding a more expressive KRS

B. A small case study of building an interoperable tool using OKBC.