Proposed Disposition: Resolved

OMG Issue No: 10881

Title: Herbrand Semantics

Source:

NIST (Mr. Conrad Bock, conrad.bock@nist.gov conradb@cme.nist.gov)

Summary:

Herbrand semantics. In 16.2.3 (More Advanced Concepts), under the XML example, the paragraph starting "In UML, there is a strict separation" is incorrect. The M0 level of UML can be real world individuals, not just software implementations (this is called an "analysis" application). Even when they are software implementations, they do not need to be specific ones, such as an SQL database manager. The last sentence is fine because of the qualification. The previous one makes it seem like the qualification is always the case. The entire next paragraph seems to also to omit the qualification, and I think can be dropped, since the presence of particular kinds of nulls in databases not relate to UML as generally applied. The last sentence of that paragraph can be used as a summary of the discussion.

Resolution:

Replace text as described below.

Revised Text:

The following is the current contents of these two paragraphs:

> In UML, there is a strict separation between the M1 and M0 levels. At the M1 level, that an association is mandatory (minimum cardinality greater than 0) is exactly the predicate [M]. Any difference between UML and OWL must come from the treatment of the model of the M1 theory at the M0 level. In practice, M0 models in UML applications tend to be ground Herbrand models implemented by something like an SQL database manager. For these cases, if we know a horse has a color, then we know what color it has. To the extent that UML tools and modeling build this expectation into products, conflict can occur when interoperating with an OWL ontology.
>
> But UML does not mandate M0 models to be Herbrand models. In particular SQL-92 supports the Null value construct, which has multiple interpretations, including “value exists but is not known.” Some years ago, CJ Date proposed a zoo of nulls with specific meanings, including “value exists but is not known,” and there have been proposals by Ray Reiter and others for databases with either existentially quantified variables in the data or which reason with the M1 theory for existentially quantified queries. It is possible for a particular application to introduce a special constant “unknown” into a class, which is treated specially by the programs. UML does not forbid an implementation of a class model in one of these ways. So there is no difference in principle between UML and OWL for properties which are declared to have minCardinality greater than 0 (and maxCardinality >= minCardinality) for a class.

Replace these two paragraphs with the following:

In UML, there is a strict separation between the M1 and M0 levels. If an association is mandatory (minimum cardinality greater than 0) it exactly matches the predicate [M]. Any difference between UML and OWL must come from the treatment of the model of the M1 theory at the M0 level. In practice, implementations derived from UML models tend to be ground Herbrand models implemented by something like an SQL database manager. For these cases, if we know a horse has a color, then we know what color it has. To the extent that UML tools and modeling build this expectation into products, conflict can occur when interoperating with an OWL ontology.

But UML does not mandate Herbrand compliance. It is possible for a particular application to introduce a special constant “unknown” into a class, and define special treatment by the programs. UML does not forbid an implementation of a class model in one of these ways. There is no difference in principle between UML and OWL for properties which are declared to have minCardinality greater than 0 (and maxCardinality >= minCardinality) for a class.

Proposed Disposition: Resolved