

Meta Modeling with ConceptBase

Manfred A. Jeusfeld, jeusfeld@uvt.nl
Tilburg University, The Netherlands

Christoph Quix, quix@informatik.rwth-aachen.de
RWTH Aachen University, Germany

ConceptBase [JGS95, JNS98] is a deductive database system with object-oriented abstraction principles. Its ability to represent information at any abstraction level makes it a powerful tool for meta modeling. The system is accompanied by a configurable graphical user interface that builds upon the logic-based features of the ConceptBase server. The system has been developed at the RWTH Aachen University, originally as a tool for managing heterogeneous design environments. It is divided into the *CBserver*, essentially a multi-user deductive database system kernel, and the *CBinterface*, a suite of graphical user interface tools to manipulate the contents of the CBserver. The system combines the following features to create a rich meta modeling environment:

Unlimited meta class hierarchies. ConceptBase can represent information at the data level (example data, traces of process executions etc.), the class level (schemas, process definitions etc.), the meta class level (constructs of modeling languages), the meta meta class level (constructs for defining modeling languages), and so forth.

Uniform object representation. All objects are represented in a uniform quadruple data structure $P(id, x, l, y)$ called *P-facts*. Attributes, specializations, instantiations are all represented as P-facts.

Logical expressions. Rules, constraints, and queries are expressed in first-order logic formulas. Internally, the system transforms them into Horn clauses interpreted by a Datalog-based evaluation machine. Logical expressions can range over any object regardless of its type (node vs. link) and its abstraction level. By relying on Datalog, ConceptBase inherits the minimal model semantics and a robust computational system for evaluating logical expressions. Additionally, ConceptBase offers active rules, aggregation functions, and user-defined functions.

ConceptBase can address most of the modeling tasks suggested for WoMM-2005. We highlight a few tasks and their solution to give an impression of the capabilities of ConceptBase for meta modeling. A full overview is given at the demonstration of the tool at WoMM-2005.

Static properties. Assignment of attributes to classes is straightforward in ConceptBase. Constraints are a particular strength of ConceptBase. Any property of a model can be made subject to constraints. Even more, frequent patterns of constraints can be stated as so-called meta-level formulas and then re-used by instantiating certain placeholders in the formula.

Geometrical figures. The set of geometrical figures is virtually unlimited. Any polygon shape can be represented as node type. Beyond that, nodes can be displayed by any type of Java component, e.g. bitmaps, complex forms containing buttons and text fields, or multimedia objects such as movies or sounds. The assignment of graphical types to objects is subject to a user-definable query. That allows to display objects in a way that highlights their role depending on the current database state. For example, ill-defined objects can be displayed in red color to indicate that they require some change.

Links between models. A particular strength of ConceptBase is its ability to manage multiple models represented in multiple modeling languages and to keep these models synchronized to each other. Essentially, each modeling language is represented as a collection of interrelated meta classes. A link between concepts of different models is by nature not different from a link between concepts in the same model. In fact, the very notion of a *model* is a user-definable concept in ConceptBase. Links from model elements to models are fully supported as well as links from any part of a model element to other parts or to complete models.

Versioning. ConceptBase maintains the transaction time with any object, i.e. any single piece of information. That allows to roll-back to a previous state by just specifying the rollback time. The user can easily define synchronization points.

Dynamic aspects. While simulation is not supported, ConceptBase can well analyze the result of the execution of a dynamic model, e.g. a petri net. Concepts like *enabled transaction*, *dead state* are expressible as ConceptBase queries. Proper simulation can be performed by tools interacting with the CBserver.

Performance. The query evaluation machine is highly optimized and can handle large models consisting of literally hundreds of thousands of concepts.

Integration. A programming interface is provided that allows external tools to communicate with a CBserver. A three-tier architecture with a request broker between an external tools and the ConceptBase server has also been tested successfully.

Process models for modeling. ConceptBase allows to define modeling methods by meta classes which represent the properties of modeling steps. Integrity constraints for models can be attached to the modeling steps where they should be checked.

ConceptBase is distributed free-of-charge for non-commercial usages. The latest release is version 6.1 with executables for Mac OS X, Windows, Linux, and Solaris. More information with selected solutions to meta modeling problems is available at the web site <http://www.conceptbase.cc>.

References

- [JGJS95] Jarke, M., Gellersdörfer, R., Jeusfeld, M. A., und Staudt, M.: ConceptBase - a deductive object base for meta data management. *J. Intell. Inf. Syst.* 4(2):167–192. 1995.
- [JJNS98] Jeusfeld, M. A., Jarke, M., Nissen, H. W., und Staudt, M.: ConceptBase – Managing Conceptual Models about Information Systems. In: *Handbook on Architectures of Information Systems*. pp. 265–285. Springer-Verlag. 1998.