Knowledge is more than Data

A Comparison of Knowledge Bases with Databases

Michael Witbrock ▸ Cycorp Inc ▸ 1/1/2007

Knowledge is more than Data

A Comparison of Knowledge Bases with Databases

Traditional relational databases and Artificial Intelligence Knowledge Bases have much in common in that both serve to store and retrieve information. However, the expressivity with which information can be represented and the flexibility with which it can be queried set true Knowledge Bases apart from more traditional data stores.[1] In databases, important domain information is often captured implicitly in their design (i.e., the selection of tables, fields, and their relationships) whereas in Knowledge Bases, most domain information is represented explicitly. Furthermore, Knowledge Bases support the specification of knowledge transformation rules and include an inference engine that determines when these rules should be applied. Using such transformations, Knowledge Bases can provide information that may be inferred from their content, in addition to simply retrieving previously stored data. These differences provide Knowledge Bases with both significantly greater representational flexibility and the ability to support a greater number and range of types of queries based on their contents.

Knowledge Bases automatically combine information to answer queries.

Information Access

A key advantage of Knowledge Bases over DBs is their use of a representation suitable for automated reasoning, revealing information that is only *implicitly* contained in what the KB stores. Databases are optimised for lookup: they efficiently *retrieve* specified subsets of data they contain, and make it available for further processing. But that processing must be specified fully in advance in the form of hand-crafted SQL queries, or by writing software code to do DB queries and process the results.

By contrast, Knowledge Bases store both data and *rules* within the same representation. The rules in a KB, when processed by an inference engine, allow the KB to find answers to queries by automatically combining stored knowledge as needed, rather than as specified in advance. For example, a KB containing simple rules for unit conversion can automatically use them, when appropriate, to calculate the total passenger load of an aircraft, in pounds, by converting and summing up weights given in kilograms, pounds and stones. In a DB, this, and all such calculations, must be painstakingly specified, in advance, by hand.

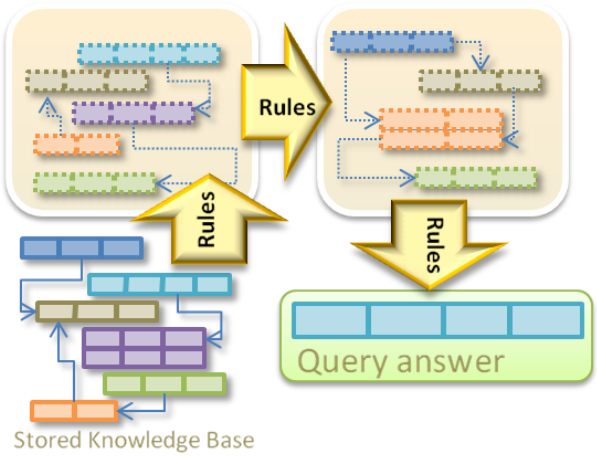

Figure 1: Databases can only use what is explicitly stored in them to answer queries. Knowledge Bases can use rules to infer additional "virtual" assertions, sometimes invoking several steps of reasoning, to obtain the data needed to produce an answer. And, since the rule logic is itself contained in the KB, when they have finished this reasoning, they can explain the rationale for each answer.

[1] The term "Knowledge Base" is sometimes used generically to refer to any repository of enterprise information including a searchable corporate intranet, wikis, databases, document repositories, etc. In this document, we use the term to describe a rich, machine-accessible information store including rules for manipulating that information.

Copyright © 2007 Cycorp, Inc.

Information Representation

Knowledge bases use a uniform representation based on logic. This representation is sufficiently powerful and general to eliminate the need to redesign the KB every time a new kind of information needs to be stored.

Databases

Databases store their content as a set of relations, organized in tables. If the form of the data to be stored is known in advance, in detail, relational databases can store and retrieve it very rapidly, and in vast quantities. However, this representation is very inflexible. Adding a new type of data to a relational database requires design of a modified schema that represents the new data in exactly the right form to interoperate with existing data. This is often difficult and may be nearly impossible. As the number of types of data that must be represented increases, schema design becomes more and more difficult.

Knowledge Bases do not need to be redesigned to add a new type of information.

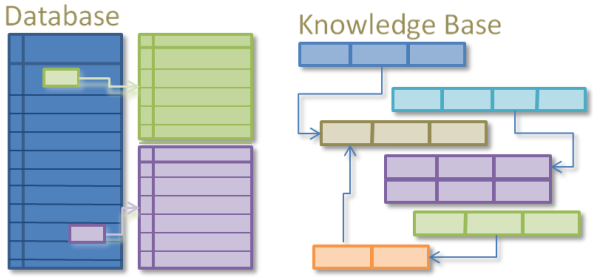

Figure 2: Databases store large amounts of data in fixed, predetermined patterns. Knowledge Bases use a flexible representation, based on logic, allowing new types of data to be added, and interrelated, on the fly.

Knowledge Bases

Knowledge Bases use a uniform, logical, representation, supported by an ontology, allowing new types of data to be described in terms of existing knowledge base content; this is far more flexible. To add a person's address to a knowledge base that has never stored an address before, one need simply *assert* that an address is a kind of location relation between a place and a person, and then that, say, George Bush's address is 1600 Pennsylvania Avenue. This process is fast and straightforward, whether the KB will eventually store millions of addresses, or just one. By contrast, in a database, one must add a new table, and ensure that the representation of George Bush (and his address) in that table is syntactically identical to all other uses in the Database.

Ontologies

When the set of facts and rules in a KB contain taxonomic information (such as that the United States is a country, and that a country is a type of geopolitical entity), they are often referred to as an ontology. The taxonomic information in an ontology is particularly useful when using the knowledge in the KB, since it allows the system to determine what types of information to include during automatic problem solving (for example, by knowing that medical doctors and lawyers are professionals, but that short-order-cooks are not, it can determine whom to include in a calculation of change in average professional income). In a DB, each different use of such information must be specified, in advance, in SQL or other computer code.

Case Study: Cyc-enhanced IMDb

The Internet Movie Database (www.imdb.com) contains a wealth of information about movies and actors, but the content of most of its fields, as well as the meaning of most of its attributes are completely opaque to the DB (and the website) itself. Thus, one can search for a movie of certain genre (say, "Horror") but not for current movies that are inappropriate for a 5-year-old child; one can find a movie set in a desert, but not find the same movie if searching for one showing sand; one might find a movie filmed in China, not one filmed in the Orient. Other searches may be more cumbersome than need be (e.g., "Find an animated movie with that guy who was in Lord of the Rings," or "Are there any Elizabethan romances playing within ten miles of my house that start within the next hour?" or "What movie star a US President and his wife?") Such queries are only possible when knowledge and reasoning extend typical database lookup capabilities.

Copyright © 2007 Cycorp, Inc.