Available online at www.sciencedirect.com
ScienceDirect
IFAC-PapersOnLine 48-4 (2015) 220–225

Ontology versus Database

*Michal Sir ** Zdenek Bradac ***
Petr Fiedler

* *Department of Control and Instrumentation, Brno University of Technology Czech Republic (Tel:+420541141308; e-mail: sir@feec.vutbr.cz).*
** *Department of Control and Instrumentation, Brno University of Technology Czech Republic (email: bradac@feec.vutbr.cz).*
*** *Department of Control and Instrumentation, Brno University of Technology Czech Republic (email: fiedlerp@stud.feec.vutbr.cz)*

Abstract: The goal of this paper is to clarify the differences between ontologies and databases. The article describes, in a step-by-step manner, the parts in which differences occur. However, there are also similarities proving that ontologies and databases are not completely different. Based on these aspects, this paper presents various approaches to transforming a database to ontology. The conclusion summarizes and highlights the most important similarities and differences.

Keywords: Ontology, database, structure, differences, similarity

1 INTRODUCTION

Some of us still remember the time when the computers were as large as a ballroom. During the period, however, most data (such as library booklists, borrowing records or lists of employers) were safe in file cabinets, which allowed data organization according to different criteria to classify a new entry. Most file cabinet operations were performed by people; data elimination was nevertheless carried out using paper shredder machines, a tool already existing at that time.

Currently, computers dominate the world, and therefore the above-discussed data types are stored in computer warehouses. Being replaced with databases, file cabinets are progressively disappearing; in recent years, databases and ontologies have been widely applied. While databases are already well known and used as a part of the everyday working routine, ontologies are gaining in popularity only gradually (despite the increasing preferences). Both of these “organized data warehouses” are, however, becoming integral elements of our lives.

1.1 Historical differences

From the historical point of view, these two domains are very different. If we intend to compare ontology and database before they became IT instruments, we have to return hundreds of years into the past. At the time of its first use, ontology was a philosophical discipline; Aristoteles worked with the concept and referred to it as “the first philosophy” (384 – 322 BC), (Sir et al. 2011). He defined ontology as the doctrine of being. Another major period in the development of the notion was the Renaissance, during which functionalist ontology was created by Nicolaus Cusanus. In modern philosophy, the concept came to be associated with rationalism, an approach that represents the union of being and thinking.

As mentioned above, the history of databases began somewhat differently; although we cannot exactly determine when file cabinets were created, this probably occurred at the time the first library attempted to register the first volume. The next stage of databases consisted in the actual transfer of filing cabinets to machines; this period could be dated to the 1890s, a time when a large census was organized in the USA. Generally, then, data were stored on punched cards, and the processing was carried out on electromechanical machines. A higher-level computer language, COBOL, appeared in the 1960s. This language was then used for many years as a significant data processing tool. In 1971, the Database Task Group (DBTG) issued a report titled *The DBTG April 1971 Report*. The report included proposals of a database scheme, a language for the scheme definition, and a subscheme. Another part of this document was a complete description of a networked database system. Around 1970, relational database was mentioned for the first time, and its inception is ascribed to E. F. Codd. The modern history of databases then began in the 1990s; it was in this decade that object-oriented databases emerged. Although these databases were originally to replace the old relational model, the attempt failed, and object-relational database was created as a compromise between relational and object-oriented databases.

From the evolutional point of view, ontology underwent more prominent development than database: it started as a philosophical discipline analyzing existence and being and gradually expanded into information technologies. Databases (or, previously, file cabinets) have always served the same purpose: today, they are used similarly as in the past. The major difference in this context consists in the changed agent or operator, namely in the move from humans to computers.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.07.036

2 BASIC DIFFERENCES

For a thorough explanation of the characteristics of database and ontology, several key terms have to be defined. The first such concept is the UNA (Unique Name Assumption); the definition of this term asserts that there is only one word available for one entity from the real world. (Poote and Mackworth, 2010)

Other concepts include primarily the CWA (Close World Assumption) and the OWA (Open World Assumption); both of these terms are used for knowledge representation (Dutra, 2009), (Matthews, 2006). The CWA is utilized by systems that comprise complete information; these are mostly database applications. For example, we can point to the database of flight companies, which enables us to find a direct flight, such as that from Prague to Vienna. If the database does not include the flight-related data, a clear result will be returned (0 or NULL), and the interpretation is that no direct flight from Prague to Vienna is available.

The OWA is applied if the system contains incomplete information. This concept represents concrete knowledge and indicates how new information can be found. To provide an example, we could refer to patient history systems in the medical field. If a given patient card does not include information about the patient and their allergies, it is not correct to say that the patient does not suffer from allergy now: we cannot establish whether the patient is treated for allergy until additional information to confirm or refute the hypothesis is found.

Generally, we can say that the CWA returns “0” (information missing) and the OWA returns “I do not know”. However, if we investigate the claim in greater detail, we find that it is not entirely true. For example, let us consider the assertion “Ivan is a citizen of the CR”. Another argument (regarded as true) is "a person can be a citizen of one country". If we juxtapose these two situations, everything is in order because Ivan is only a citizen of the CR. If we then have the argument “Ivan is a citizen of the SR”, an error occurs in the CWA system because we mentioned that the person has to be a citizen of only one state. In the given case, the OWA does not generate an error; this system will nevertheless produce the following claim: “If Ivan is a citizen of one state and a citizen of the CR and the SR, then the CR and the SR must be same”. The OWA returns this answer because it uses the UNA. The CWA includes this property but the OWA does not. The system OWA does not use UNA, yet the UNA can be added to the system artificially. The principle is to add disjunction to individual names: in other words, we determine the differences of names.

The main difference between ontologies and databases lies between the OWA and the CWA. From the examples mentioned above, it can be inferred that while ontologies utilize the OWA system of knowledge representation, the CWA is used by databases. A database exploits the UNA for naming entities. Any information missing in a database system has the value of “0”. Any item of information missing in an ontology system is considered unknown.

3 METHOD OF CREATION AND PURPOSE

Another major difference between ontologies and databases is the purpose for which they are created. While ontologies are focused on adding meaning and comprehension, databases concentrate on data storage. In other words, databases are created as effective data warehouses, whereas ontologies are formed for better communication, interoperability, and as the communication bridge between a human and a machine.

Both of these systems use different creation methods. A database system is created from scratch, which means that all tables and their contents are designed in manner indicated here. When we design an ontology system, we attempt to take advantage of existing ontologies or system structuring upon an existing ontology (we thus extend an existing ontology). For example, the SSN ontology is built on the DOLCE upper ontology.

In creating a database system, we apply the normalization of tables; such normalization is used to delete redundant data from the tables and to reduce the complexity. For the given purpose, normal forms are used in the database system; the forms are a set of rules that help to correct the transformation of entities and relationships to the structure of the physical layout of the tables. Today there exist six normal forms, but only the first three are used.

The methodology for the creation of ontology does not include normal forms. A major ontology creation method consists in design patterns. These patterns, however, are not as strict as normal forms: rather than that, they create general rules. In this context, the author recognizes six areas (Ontology patterns 2014):

- Structural pattern
- Syntactic pattern
- Content pattern
- Presentation pattern
- Consideration pattern
- Corresponding pattern

3.1 *Ontology creation*

Ontology creation is a process comprising several stages. Some of these phases, namely specification, conceptualization, implementation, and maintenance are materialized during the development process; other operations are performed throughout the entire existence of the ontology: data mining, documentation, and evaluation. Some of the main procedural imperatives are described below.

Identify the integration possibility: The framework being applied to build the ontology should allow for some kind of knowledge reuse. In certain cases, integration may involve rebuilding the ontology in a framework different from that where the ontology is available. In some situations, this may be cost-effective, but in others it could be more profitable to