Chapter 1.22
Knowledge Representation

Gian Piero Zarri
University of Paris IV/Sorbonne, France

INTRODUCTION

In 1982, Allen Newell introduced the "knowledge level" principle (Newell, 1982) and revolutionized the traditional way of conceiving the relationships between knowledge management and computer science. According to this principle, the knowledge level represents the highest level in the description of any structured system: Situated above and independent from the "symbol level," it describes the observed behaviour of the system as a function of the knowledge employed, and independently of the way this knowledge is eventually represented/implemented at the symbol level. "The knowledge level permits predicting and understanding behaviour without having an operational model of the processing that is actually being done by the agent" (Newell, 1982, p. 108). An arbitrary system is then interpreted as a rational agent that interacts with its environment in order to attain, according to the knowledge it owns, a given goal in the best way; from a strict knowledge level point of view, this system is then considered as a sort of "black box" to be modeled on the basis of its input/output behaviour, without making any hypothesis on its internal structure. To sum up, the knowledge level principle emphasises the why (i.e., the goals), and the what (i.e., the different tasks to be accomplished and the domain knowledge) more than the how (i.e., the way of implementing these tasks and of putting this domain knowledge to use).

BACKGROUND

The emergence of the knowledge principle produced a shift of emphasis, in the (computerized) knowledge management domain, from a pure "representational" attitude to a "modeling" one, that is, a shift from the production of tools for implementing the knowledge a system will use to that of tools for building up models of the behaviour of the system in terms of that knowledge. An example of this is the Knowledge Acquisition and Design Structuring (KADS) methodology (Schreiber, Wielinga, & Breuker, 1993; Schreiber, Akkermans, Anjewierden, de Hoog, Shadbolt, Van de Velde, & Wielinga, 1999). A fundamental step in the KADS approach is, in fact, the set up of a general "conceptual model" of the system, which an observer (a knowledge engineer) creates by abstracting from the problem-solving behaviour of some experts. According to the knowledge principle, the conceptual model does not include any detailed constraint about the implementation level. This last function is assigned to the "design model," which corresponds to the (high level) specifications of the final knowledge-based system (KBS), and which represents the transformations to be executed on the conceptual model when we take into account the external requirements (e.g., specialised interfaces, explanation modules, etc.). The conceptual model is built up according to a four-layer structured approach: Each successive layer interprets the description given at the lower layer. A first layer concerns the "static domain knowledge," that is, the domain concepts and their attributes, the domain facts, the structures representing complex relations, and so forth. The static knowledge can be viewed as a declarative theory of the domain. A second type of knowledge concerns the "knowledge sources" and the "metaclasses." A knowledge source is defined as an elementary step in the reasoning process (an inference) that derives new information from the existing one; KADS presupposes the existence of a set of canonical inferences such as "abstraction, association, refinement, transformation, selection, computation." Metaclasses describe the role that a group of concepts plays in the reasoning process (e.g., observable, hypothesis, solution, etc.). The third layer contains knowledge describing how inferences can be combined to fulfill a certain goal, that is, how to achieve operations on metaclasses. The most important type of knowledge in this category is the "task": A task is a description of a problem-solving goal or subgoal, as "diagnose a patient with these particular symptoms." The fourth category of knowledge is the "strategic knowledge," which settles the general goals that are relevant to solve a particular problem; how each goal can be achieved is determined by the task knowledge.

One of the main attractions of this structured, analytical approach to the automation of knowledge management resides in the fact that all the methodologies based implicitly or explicitly on the knowledge level principle embrace the idea that the set up of KBSs can be facilitated by the development of libraries of reusable components. These pertain mainly to two different classes, (1) reusable "ontologies" (see also Zarri, "RDF and OWL" in this Volume) and (2) reusable problem-solving methods, which define classes of operations for problem-solving. Chandrasekaran (1990) is one of the first to suggest the development of reusable components under the form of "generic tasks," where a generic task defines both a class of application tasks with common features, and a method for performing these tasks.

An additional manifestation of this general tendency toward generalisation, abstraction, and reuse in the knowledge management domain are the activities aimed at the construction of general and reusable "corporate memories," (see van Heijst, van der Spek, & Kruizinga, 1996; Brooking, 1998; Beckett, 2000). Knowledge has been recognised as one of the most important assets of an enterprise and a possible success factor for any industrial organization, on the condition that it can be controlled, shared, and reused in an effective way. Accordingly, the core of the organization can then be conceived under the form of a general and shared organizational memory, that is, of an online, computer-based storehouse of expertise, experience, and documentation about all the strategic aspects of the organization. The construction and practical use of corporate memories becomes then the main activity in the knowledge management of a company, a focal point where several computer science and artificial intelligence disciplines converge: knowledge acquisition (and learning), data warehouses, database management, information retrieval, data mining, case-based reasoning, decision support