Representation Levels Within Knowledge Representation

Heather D. Pfeiffer and Joseph J. Pfeiffer, Jr.

New Mexico State University
emails: hdp@cs.nmsu.edu and pfeiffer@cs.nmsu.edu

Abstract Representation of knowledge is used to store and retrieve informational data in a machine. Since *meaning* cannot be directly stored in the computer; this work proposes a series of levels of representation. The meaning of the data is transformed to a format that the machine can use to store and retrieve knowledge. These levels are designed to transform the knowledge from an abstract definition to a machine representation without loosing any meaning.

1 Introduction

Knowledge gives a definition or understanding of events and acts within the world; knowledge describes the world and gives it *meaning*. For the computer the description of the problem that it is to solve has become known as *knowledge representation (KR)*. The representation consists of a set of syntactic and semantic rules to describe a problem domain [1]. KR, when abstractly described as conceptual ideas or in natural language, appears very informal and without concrete machine structure.

Some of the confusion in the field of knowledge representation is what rules, syntactic or semantic, are defined when looking at an idea with an informal representation and then with a machine processable representation. In many readings, it is not made clear what knowledge can be processed directly by the computer as machine code, and what must be transformed (mapped) from another more abstract representation. It should be noted that, in general, abstract language representations are too informal for machine processing; therefore most knowledge representations must be translated to a more concrete representation using concrete structural languages in order to be coded. Then execution and analysis can be performed.

2 Background

Back in 1971, Shapiro [2] attempted to divide all representations defined by semantic networks into the following two levels: 1) item - conceptual level and 2) system - structural level. Levelization only looked at the actual semantic network represented on the page, and did not consider how to code for machine processing. The item level was concerned with the nodes that appeared in the network, and the system level attempted to define the links that were present between the nodes in the network.

In 1979, Brachman [3] tried to address the confusion about representations of knowledge by defining levels for different types of representations. Brachman described one representation in terms of another; however, when levels are defined by other levels and representations are defined by other representations there is still confusion [4]. In his paper, Brachman defines a "level" as a distinctive type of network of nodes or links, and gives the following levels: 1) implementation level, 2) logical level, 3) epistemological level, 4) conceptual level, and 5) linguistic level.

For the five levels given above, Brachman saw the implementation as the lowest level; that is, the most basic type of network. The epistemological level is seen by Brachman as a missing level, located between the logical level and the conceptual level, which in a network links formal structure to conceptual units and creates a set of their interrelationships. Guarino, like Brachman, also saw missing information in the levels, and added an ontological level to Brachman's classification levels. The ontological level would give a foundation for the knowledge engineering process and depict a set of features for the computational properties of each level[5]. For Brachman and Guarino, all the levels are processed as part of the knowledge representation.

Brachman did not try to actually look at processing representation from a computer processing point of view. Then in 1982, Newell [6] began the redefinition of a "level" as needed for computer processing. He defined a level in the following way: "a level consists of a medium that is to be processed, components that provide primitive processing, laws of composition that permit components to be assembled into systems, and laws of behavior that determine how system behavior depends on the component behavior and the structure of the system" [6]. Newell referred to computer systems levels as the following bottom (highest) to top (lowest) sequence: device level, circuit level, logic level, register-transfer and program level, and configuration level. Therefore within his levels, Newell renamed the program level to symbol and added a new level just after that known as the knowledge level.

3 Representation Levels

This work expands on Newell's computer systems level idea, in particular investigating what could be the possible computational mechanisms or physical structures of the symbol level (representations), while seeing level relationships more from Brachman's definition [3]. That is: "there is a *level* of processing of representations that sees the lowest level to be a very abstract representation and then, as levels increase, the representation becomes more concrete or machine like" [3]. The highest level of representation would be processed directly by a computer (see Figure 1 [7]) because it is the actual implementation that is compiled or interpreted as machine code.

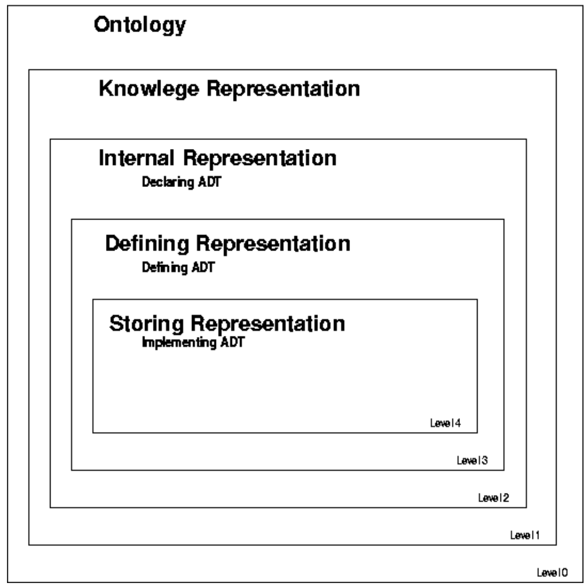

Figure 1. Levels of Representations

Consider representation in a system to be a series of processing levels. Encapsulating the KR is the level of *ontological information* [8], level 0, with general hierarchy information[9]. This level would be considered the knowledge level under Newell's levels, part of the linguistic level for Brachman, and would be a relocation of Guarino's ontological level. The information represented is not actually part of the structure of the domain knowledge and is the most abstract of all the levels. In fact, it is more of a hierarchy of conceptual information than knowledge, so it will be called "ontology" [10].

KRs will start processing at level 1. For Newell, the *knowledge representation* level would be part of the symbol level, very close to the knowledge level. In Brachman's levels this would encompass part of the conceptual level and all of the epistemological level. Level 1 is the first real translation from a conceptual idea to actually being able to represent the *concept* to a computer.

The second level of representation, level 2, is an *internal representation* that could be viewed as a virtual machine. This is where the declaration of an abstract data type (ADT) is performed. This syntactic representation is more formal and can be used in the definition and implementation of the declared ADT. The syntactic rules are concrete and define a mapping of symbols to operators. However, in order to implement this level, there must be a third level of definitions giving more structure.

Level 3 consists of the actual *semantic definition* of the ADT declared in level 2. The semantic rules are concrete, and define a mapping of operations to functions. It defines the algorithms to be performed, and theoretical time/space analysis can be performed on these algorithms. There is a strong connection between level 2 and level 3 because the concrete rules of the representation in level 2 will work over the algorithms of level 3 during the implementation of the data structures at the next level.

The innermost level of representation, level 4, is the *actual implementation* of the ADT definition. This level is where all the data structures come together. It is at this level that a computer language, or a newly defined language is chosen [11]. The coding of data structures and algorithms will be performed, and the representation is the most concrete.

4 Conclusion

Each of the levels of representation defined here move through three categories of data: 1) meta-data, 2) abstract data, and 3) concrete data. The computer is able to process information within the concrete area even though people actually deal with most information under the abstract or meta-data area. By a knowledge engineer breaking down conceptual ideas at different representation levels, they can see how to transform knowledge from data seen by humans to data that can be processed by machines without loosing information and structure. Level 0 of representation holds meta-data to help in preventing any loss of knowledge.

References

1. Aidinejad, H.: Semantic networks as a unified model of knowledge representation. MCCS-88-117 (1988)
2. Shapiro, S.C.: A net structure for semantic information storage, deduction, and retrieval. In: Proceedings of the 2nd International Conference on Artifical Intelligence. (1971) 512–523
3. Brachman, R.: On the epistemological status of semantic networks. In Findler, N., ed.: Associative Networks: Representation and Use of Knowledge by Computers. Academic Press, New York (1979) 3–50
4. Charniak, E., McDermott, D.: Introduction To Artifical Intelligence. Addison-Wesley, Reading, MA (1985)
5. Guarino, N.: The Ontological Level. In: Philosophy and the Cognitive Science. Holder-Pivhler-Tempsky, Vienna (1994) 443–456
6. Newell, A.: The knowledge level. Artifical Intelligence **18**(1) (1982) 87–127
7. Pfeiffer, H.D.: The Effect of Data Structures Modifications On Algorithms for Reasoning Operations. PhD thesis, New Mexico State University, Las Cruces, NM (2007)
8. Oehrstoem, P., Andersen, J., Scharfe, H.: What has happened to ontology. In Dau, F., Mugnier, M.L., Stumme, G., eds.: Conceptual Structures: Common Semantics for Sharing Knowledge. Volume 3596 of LNAI., ICCS2005, Springer (July 2005) 425 – 438
9. Lehmann, F., ed.: Semantics Networks. Pergamon Press, Oxford, ENGLAND (1992)
10. Sowa, J.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA (1984)
11. Welty, C.A.: In Integrated Representation for Software Development and Discovery. PhD thesis, Vassar College (1995)