İÇİNDEKİLER

1. INTRODUCTION

In this paper, I aimed that to explain the process of the conversion data to wisdom clearly. Also in this process I included additional information about data, knowledge, information and wisdom. All these are really essential to understand knowledge base systems clearly. After that I explain classification knowledge representation methods and drew a figure for every method. With this study, every body can understand what is knowledge and knowledge based systems and where it can be used.

2. WHAT IS KNOWLEDGE?

The definition of knowledge is widely. In the context of decision support, we have defined knowledge as a progression (Figure 1). In a progression we begin with data which is of limited utility. By organizing or analyzing the data, we understand what the data means, and this becomes information. Understanding the meaning of the data (information) in the context of interactions, relations, or problems allows for the interpretation or evaluation of the data to yield knowledge or prediction. Finally, the application of a decision process and decision criteria leads to an action or a decision.

Figure 1: Knowledge Progression

Gene Bellinger, in his Outsights "Mental Model Musing" web page, uses a slightly different definition. His concept is that as the degree of "connectedness" and understanding increase, we progress from data through information and knowledge to wisdom (Figure 2). While we may argue that understanding principles may encompass knowledge rather than wisdom, we will avoid the issue of what is truth altogether.

Figure 2 : Bellinger's Knowledge Model

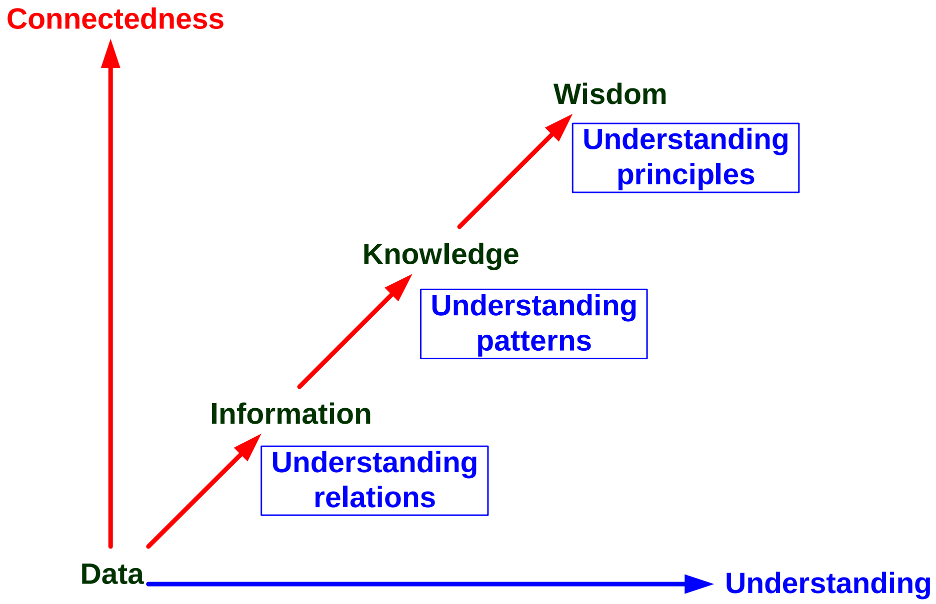
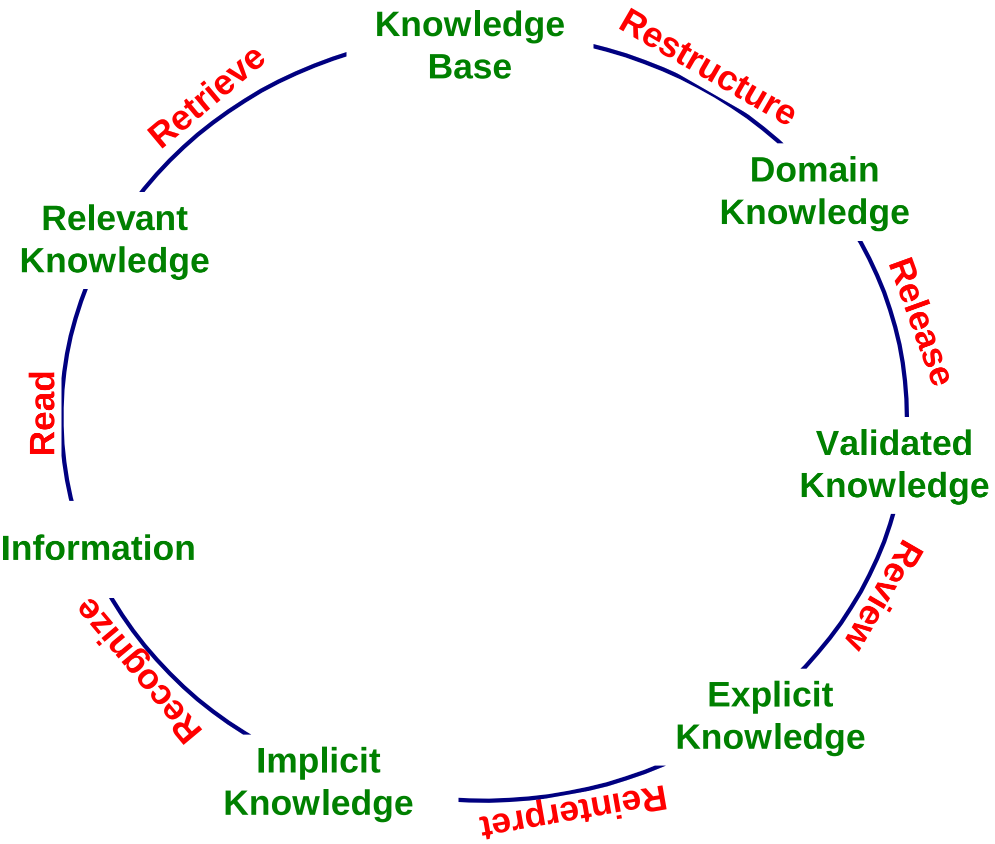


Figure 3: Knowledge Cycle



Butcher and Rowley (1998) identified a counter-clockwise knowledge cycle (Figure 3). In their paper, they identified the seven R's of knowledge management:

1– **Retrieval** delivers relevant information or knowledge from the corporate knowledge base or from external sources.

2– **Reading** the knowledge retrieved provides information to the reader.

3 –**Recognition** internalizing the information places it in the context of a new problem, as well as the readers experience to create implicit knowledge.

4 – **Reinterpretation** in the new context and formalizing implicit knowledge converts implicit knowledge to explicit knowledge.

5 –**Reviewing** peer or operational review of the new explicit knowledge produces validated knowledge.

6 – **Release** of the validated knowledge adds to domain knowledge.

7 – **Restructuring** the domain knowledge allows for its incorporation into the knowledge base.

Our use of the terms *implicit* and *explicit* knowledge generally corresponds to as tacit and explicit knowledge or as informal and formal knowledge. This discussion is designed to convey the impression that knowledge is both more complex and useful than information or data and that it is strongly dependent on context. This complexity imposes a burden on information and knowledge management systems to provide the context that defines the information and knowledge. Thus, in the same sense that we require meta data to understand data, we will also require meta information and meta knowledge to understand and appropriately apply information and knowledge. The knowledge cycle also represents knowledge as an accretionary process with knowledge building on knowledge.

3. TYPES OF KNOWLEDGE

In language, propositions can be distinguished from one another according to their methods of verification, their function, grammatical labels that can be meaningfully attached to them, and by comparisons and contrasts with prototypical examples.

From this perspective, knowledge can be organized into the following seven categories:

1) **Logical Knowledge**
2) **Formal Knowledge**
3) **Mathematical Knowledge**
4) **Grammatical Knowledge**
5) **Theoretical-Hypothetical-Empirical Knowledge**
6) **Historical Knowledge**
7) **Factual Knowledge.**

A set of logical and linguistic criteria is used in this categorization.

3.1. Logical Knowledge

Logical knowledge includes the rules of logic, logical function definitions and the logical relationships between concepts in language, and is totally domain independent. Logical knowledge is provable only in the logical grammar of language; empirical methods of verification such as observation and experimentation do not apply to it. The prefixes “By definition,” and “According to the axioms,” are meaningful for logical propositions, while the prefixes “According to the hypothesis,” “Probably,” and “According to (so and so)” are not.

3.2. Formal Knowledge

Formal knowledge includes concept definitions and member-class and class-superclass relationships. Formal statements are not verified by empirical methods; their truth and falsity is implied by the conceptual structure of the language in which they are expressed. The prefix “By definition” is applicable to formal statements, but not the prefixes “According to hypothesis,” “Probably,” and “Possibly”. Formal knowledge is less abstract than logical knowledge, for some formal knowledge can be domain dependent.

3.3. Mathematical Knowledge

Mathematical knowledge involves symbolic and numerical relationships in language. In principle, mathematical statements are verified by logical, formal and mathematical methods, but not by empirical methods.

3.4. Grammatical Knowledge

Grammatical knowledge involves meta statements about concepts and expressions in language. They are not verifiable by logical, formal or empirical methods, but accepted as true by a set of values in language (or the “deep grammar” of language), or by intuition.

3.5. Theoretical/Hypothetical/Empirical Knowledge

Theoretical/hypothetical/empirical knowledge involves general statements about natural phenomena. Statements of this category are tested against such phenomena. They are essentially verifiable (or falsifiable) by empirical methods. The prefix “By definition,” is not meaningful for these statements, while the prefixes “According to the theory,” and

"According to the hypothesis," are meaningful. Generalizations and specializations are applicable to statements in this category.

3.6. Historical Knowledge

Historical knowledge includes the records of past experience. The statements in this category are verified or falsified by the methods of historical study.

3.7. Factual Knowledge

Factual knowledge on the other hand, reflects the current state of affairs about the world. Factual statements are verified by observation. They describe specific facts or events, and therefore, specialization cannot be applied to them. Categorization of descriptive knowledge provides clarity in the design, development, operations, and evaluation of complex knowledge systems.

4. KNOWLEDGE FUNCTIONS

There are four functions associated with knowledge and knowledge management. These are knowledge acquisition, knowledge representation, knowledge management, and knowledge usage. These functions represent a logical sequence but are not necessarily formalized. For example, informal knowledge acquisition and usage may occur without formal knowledge structures or knowledge management procedures. The following discussion will introduce these four functions to establish context.

4.1. Knowledge Acquisition

Knowledge acquisition is the identification of relevant information sources and transformation of that information into knowledge. It involves at least the three activities described below.

4.1.1. Developing new knowledge (research)

Formal research programs are directed towards the development of new information and knowledge. These may be scientific research, market research, historical research, *et cetera*.

4.1.2. Adding value to information to create knowledge

The evaluation and analysis of public and corporate data to identify patterns and relationships can create valuable knowledge. The highly promoted field of data warehousing and data mining are an increasingly significant component of knowledge creation. The product of this process may be implicit or explicit knowledge.

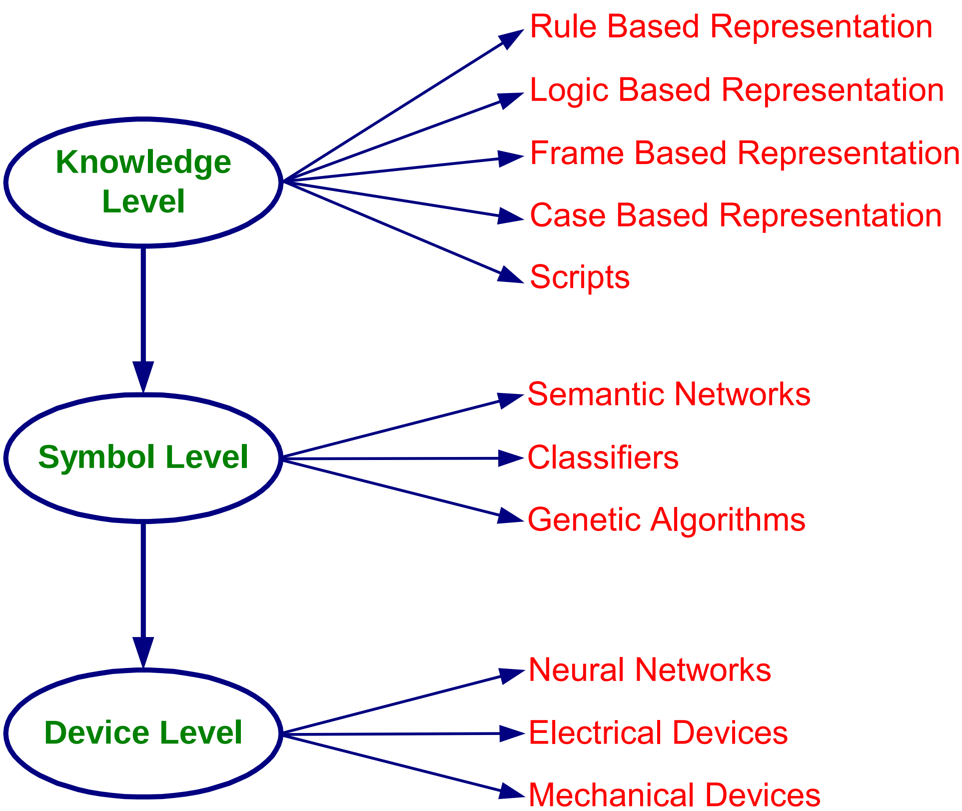
4.2. Converting Implicit to Explicit

Human experience, without recourse to rigorous scientific procedures, often generates valuable knowledge. The term *tacit knowledge*, coined by Polanyi, 1977, refers to knowledge that is the result of the integration of personal experience, beliefs, perspectives, and values. While we call it implicit knowledge, it is the application of personal experience and beliefs with data and information from similar contexts for application to a new problem or issue at hand. This informal, tacit, or implicit knowledge and its conversion to explicit or documented knowledge is a major area of information research and is the principal target for many belief matrix and rule-based expert systems.

4.2.1. Knowledge Representation

Knowledge representation allows the storage of knowledge in structures that enable the fast and accurate access of knowledge as well as the easy understanding of the content and its structure (Brachman and Levesque 1985, Brachman *et al* 1992). The conversion of implicit or informal knowledge to explicit or formal knowledge requires the creation of formal knowledge representations. These may be documents, diagrams, flow charts, databases, *et cetera*. The electronic storage and retrieval of knowledge requires the creation of formal knowledge structures such as anthologies, data models and schemas, semantic networks, *et cetera*.

Figure 4: Classification of knowledge representation methods by representation level



4.2.1.1. Rule Based Representation

Rule based representation includes the following basic elements:

- **Rules**
- **Facts**
- **Interpreters**
- **Translators**
- **Explanations**
- **Explanation functions**

Many rule based systems allow for multiple representations of rules, one representation might be for data entry, another for explanations. Explanations are usually generated by translating the rules taking part in the decision into natural language. In rule based systems, knowledge is represented as simple and compound propositions and condition-action rules. These systems can use both forward and backward chaining. In addition to its memory of facts and rules, a rule based system uses a working memory (or dynamic memory) to store temporary assertions. Rule based systems have been helpful to refine and clarify existing knowledge in their domain. They have proved invaluable as a practical means for transforming poorly understood knowledge into a coherent knowledge base. In contrast to conventional programming, rule based programming requires one to think more analytically than procedurally.

Figure 5: Knowledge representation by rule based representation method about planets

Facts :

Mercury is a planet.
Venus is a planet.
. . .
Venus doesn't have a satellite.
Venus has atmosphere.
. . .
Jupiter is a planet.

Rules :

If X is a planet and X has a satellite,
then X has atmosphere.
If X is a planet and X has ring,
then x has a satellite

Rule based systems owe their success to a small search space of a specialized knowledge-intensive system, but they have shortcomings. Hayes-Roth (1985) identifies these as follows:

1) Lack of a precise analytic foundation for deciding which problems are solvable.
2) Lack of a suitable verification methodology or a technique for testing the consistency of a rule set.
3) Absence of a system of knowledge organization that would facilitate scaling up without loss of intelligibility of performance.
4) Absence of well-developed rule compilers.
5) Lack of methods for easy integration into conventional data processing systems.

Currently, there is not a formalism to classify facts and rules in rule based systems. As they are essentially knowledge level systems rule based systems allow the implementation of deductive and inductive learning methods such as specialization, generalization and abstraction. By themselves, rule based systems cannot represent structured knowledge such as can be represented by frames. Unlike predicate logic based systems, they rely on simple inference mechanisms. Many rule based systems combine frame or logic based features. Some large rule based systems have the blackboard control architecture. In these systems, rules are organized into “knowledge sources”, which send their output to a “blackboard”. A scheduler controls the activities of the knowledge sources in accordance with the problem states indicated by the blackboard.

4.2.1.2. Frame Based Representation

Frame representation is based on the theory of frames developed by Minsky (1977). In these systems knowledge is represented in data structures called “frames” which can incorporate sets of attribute descriptions called **“slots”.** A frame system provides constructs for organizing objects and classes in taxonomic structures. In this way, frames provide structured representations of objects or classes of objects, so that a frame can contain prototype descriptions of members of a class, as well as descriptions of the class as a whole. These prototypes can be used in creating default descriptions of the objects.

Figure 6: Knowledge representation in frame based systems

Frame : Jupiter

Class :	**planet**
Distance to sun:	**V1**
Diameter :	**V2**
Volume :	**V3**
Mass :	**V4**
Orbit speed :	**V5**
Rotation speed :	**V6**
Year :	**V7**
Ring system :	**V8**

Frame systems are particularly powerful in inheritance based inference, because the taxonomic relationships among frames enable descriptive information to be shared among multiple frames by inheritance. Frames also facilitate reasoning about events and processes, and consequently the systematic implementation of theory revision methods. Since frames allow the structured representation of descriptive knowledge, they also facilitate analogical reasoning.

Frame systems provide no direct methods for describing how the knowledge stored in frames is to be used. Associations of domain dependent behavior with frames are made by attaching to them procedures written in the host programming language.

There are basically three types of procedures which apply to the data in a frame: constraints, demons and watchdogs.

- **Constraints** are activated when a slot is to be updated, to make sure that erroneous or irrelevant information is not placed into the slot.
- **Demons** are fired after an update of information in a slot takes place, and they execute a series of commands.
- **Watchdogs** are activated whenever the value in a slot they are attached to is accessed rather than updated.

The effects of the constraints, demons and watchdogs depend on the application. Each procedure is attached to a specific slot related with the data in that slot. These procedures can be inherited by other frames.

Frame systems generally incorporate rule based or logic based methods to provide the necessary, effective inference mechanisms. The procedures attached to slots can be regarded as knowledge level or symbol level depending on whether they are condition-action rules or procedures. Lenat’s CYC system integrates frames with predicate logic for the efficient implementation and control of different inference mechanisms.

4.2.1.3. Logic Based Representation

In logic based systems, knowledge is represented as simple and complex predicate statements. Such systems are based on the first-order predicate calculus. This calculus provides expressive power and a well-developed formalism.

Predicate logic had been used in representing high-level concepts like infinity, continuity and causality, and also in developing various axiom systems in mathematics, physics and biology. Logic based systems rely on resolution theorem proving for their inference mechanisms. In these systems prescriptive knowledge is represented in the form of condition-action rules or procedures. The adoption of Prolog as a logic programming language by the Japanese Fifth Generation Computing Systems project as its core language has contributed the consolidation of this language in scientific and technical programming. Developments in metalogical and in parallel programming further increases the importance of Prolog.

Figure 7: Knowledge representation in logic based systems

```
Facts :
      planet(mercury).
      planet(venus).
            . . .
      no_satellite(venus).
      has_atmosphere(venus).
            . . .
      planet(jupiter).

Rules :
      have_atmosphere(X)
                  planet(X),
                  has_satellite(X).

      has_satellite(X)
                  planet(X),
                  has_ring(X).
```

Logic based systems support a variety of deductive learning methods such as explanation based learning and conceptual clustering, and inductive methods like similarity based learning and other forms of generalizations. They also support abstraction, abduction, non-monotonic reasoning and theory revision.

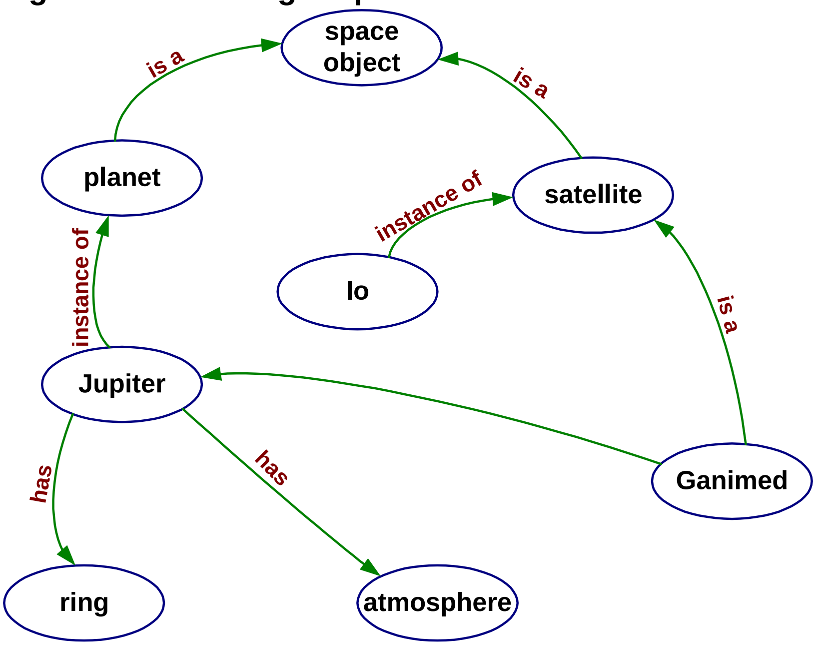
Yet, the predicate calculus itself does not have facilities for defining complex constructs such as can be expressed in frames. For this reason, purely logic based systems are less suitable for analogical reasoning than frame systems. Similarly, taxonomic and inheritance based reasoning can be implemented more efficiently in frame systems, even though they can also be implemented in logic based systems by using metalogical constructs.

Descriptive knowledge represented in frames can be translated into predicate logic statements. But, in these translations the external structure of the knowledge represented by a frame is usually lost. Frost (1986) describes how inheritance and taxonomic relationships, slot constraints and matching can be expressed in predicate logic notation. He also describes how default values can be accommodated by incorporating nonmonotonic logic.

4.2.1.4. Semantic Networks

Semantic networks (also called associative networks), are another major form of knowledge representation. In such systems, knowledge is represented in the form of nodes and arcs, where nodes represent objects, classes, and events while arcs represent relationships that hold between the concept nodes. There is a strong structural parallelism between the semantic network and predicate logic representation. Knowledge that can be expressed by predicate statements can be represented in semantic networks, with arcs representing the predicates and nodes representing terms. Semantic networks provide a scheme of pointers and back pointers that facilitate accessing information easily. In semantic networks, descriptive knowledge can be organized into concept hierarchies, which can be used to transfer inheritance properties and relations.

Figure 8: Knowledge representation in semantic networks



Examples of such hierarchies are "is-a" and "instance-of" hierarchies. "Instance-of" designates the membership of an individual to some class, while "is-a" says that one class is a subclass of another.

Yet, in semantic networks links between the concepts do not allow for exceptions as in frame systems.

4.2.1.5. Knowledge Representation in Classifiers

Classifiers (Nilsson, 1965; Hunt, 1975) and classifier systems based on genetic algorithms (Holland, 1986) are inductive learning systems. In these systems knowledge is represented as vectors or matrices of values to a set of parameters or a set of message lists.

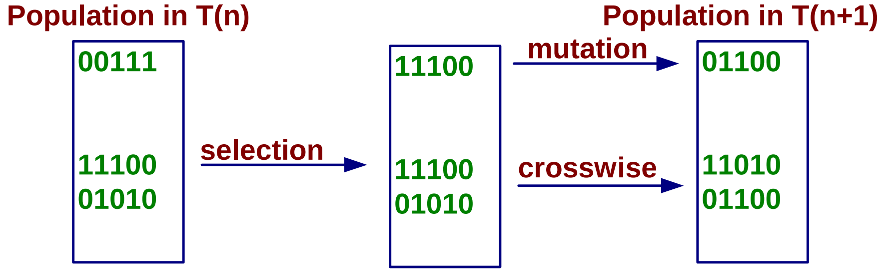
Figure 9: Knowledge representation in classifiers

Attributes	Objects		
	Bird	**Plane**	**Glider**
Has feather	**1**	**0**	**0**
Has wings	**1**	**1**	**1**
Has engine	**0**	**1**	**0**
	2	**2**	**1**

4.2.1.6. Genetic Algorithms

Genetic algorithms represent knowledge as vectors of values to a set of parameters or a set of message lists. GA based classifiers systems support, besides inductive learning, conflict resolution, classification and learning control rules.

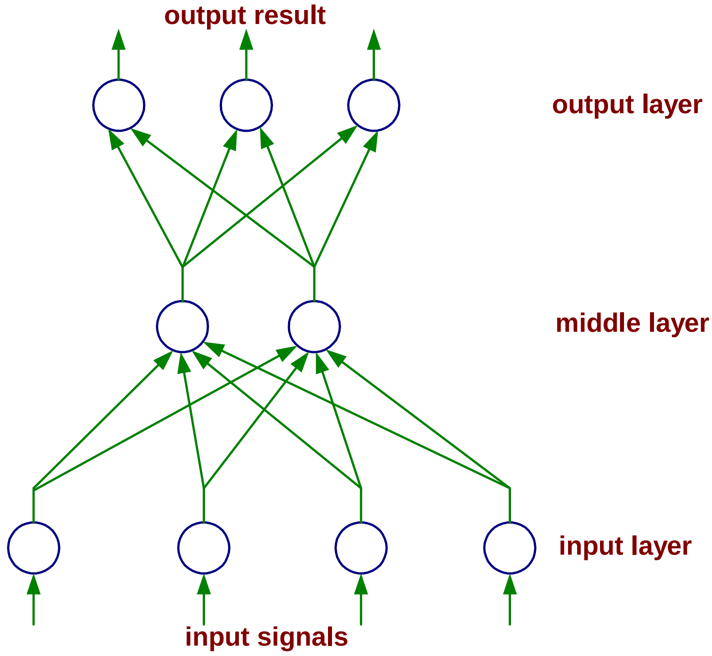
Figure 10: Knowledge representation in genetic algorithms



4.2.1.7. Knowledge Representation in Neural Networks

Neural nets are composed of a large number of interconnected processing units. In these systems knowledge is represented by the totality of the connections. Neural networks support inductive learning, classification and conflict resolution.

Figure 11: A typical artificial neural network structure



A good system for knowledge representation in a particular domain should possess the following four features:

- **Representational Adequacy** — the ability to represent all of the kinds of knowledge that are needed in that domain.
- **Inferential Adequacy** — the ability to manipulate the representational structures in such a way that as to derive new structures corresponding to new knowledge inferred from old.
- **Inferential Efficiency** — the ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanisms in the most promising directions.
- **Acquisitional Efficiency** – the ability to acquire new information easily. The simplest case involves direct insertion, by a person, of new knowledge into the database. Ideally, the program itself would be able to control knowledge acqusition

The main purposes of representing knowledge in a computational system can be summarized as follows:

- To generate and incorporate new knowledge by various methods of learning and discovery,
- To facilitate the use of existing knowledge,
- To provide explanations and facilitate the transmission of knowledge.

The basic criteria for the efficiency of knowledge representation can be listed as

- **expressive power**
- **generality**
- **extendibility**
- **flexibility**
- **understandability**
- **accessibility**

Accordingly, a good representation system must have computational efficiency, be readable by the human user, have a structure of systematic growth and improvement of knowledge, and must be applicable to various fields of knowledge. These requirements create important problems in representing knowledge in complex systems. Two of these problems are **brittleness** (see, e.g. Holland, 1986) and the **knowledge acquisition bottleneck** (see, e.g. Lenat, Prakash & Shepherd, 1986).

- **"Brittleness"** refers to the difficulties in having the knowledge system expand beyond its original scope.
- **"Knowledge acquisition bottleneck",** on the other hand, is related to the difficulties in the acquisition and representation of new knowledge.

4.3. Knowledge Management

Knowledge management should allow for simple, yet powerful access to knowledge, and its update and usage (via reasoning). It should take into account differences among users and among requirements for different problem-solving tasks. Knowledge management, like data and information management consists of the following operations:

- **Create** – the entry of additional knowledge into the knowledge base. Creation in this sense does not refer to the development or acquisition of new knowledge. It refers rather to its entry into a knowledge base.
- **Update** – the modification of knowledge resident in the knowledge base.
- **Retrieve** – the retrieval of relevant knowledge from the knowledge base for application.
- **Delete** – the deletion or elimination of knowledge from the knowledge base.
- **Archive** – the long-term storage of knowledge in a separate location from the active knowledge base. Archiving is done for security, for increased performance of the active knowledge base through the removal of no-longer-needed knowledge, or both.

The key to effective knowledge management is the representational model used to store and access knowledge in the knowledge base.

5. KAYNAKLAR

[1]. "Yapay Zekaya Giriş" Ders Notları, Şakir Kocabaş, 2002
[2]. The Management of Expertise: Knowledge, Power and the Economics of Expert Labour, Fleck J. & Tierney M., PICT, University of Edinburgh, Edinburgh, 1991.
[3] Knowledge Representtaion and Learning, Şakir Kocabaş
[4] Artificial Intelligence, Elaime Rich, Kevin Knight, 1991
[5] Expert Systems for Expert, Kamran Parsaye, Mark Chignell