UDC 004.048

*A.B.M. Salem, PhD (Ain Shams University, Egypt)*
*V. Shendryk, PhD (Sumy State University, Ukraine)*

## The Knowledge Computing and Engineering in the representation of knowledge in industrial information systems

*The overview of some of knowledge computing techniques and approaches and their applications in the representation of knowledge in industrial information systems is proposed. The paper discusses the mainly deals with techniques: Knowledge Engineering and Reasoning Techniques.*

Knowledge is the aspiring elementary resource mandatorily required by all intelligent information processing systems. Knowledge engineering is an intelligent process by which the gathered raw data is transformed into knowledge. Knowledge engineers use artificial intelligence concepts and techniques in developing knowledge-based systems [1, 2]. They have been proved to be effective and efficient in developing intelligent and knowledge-based systems for many tasks in the field of industrial information management [3, 4].

The main purpose of this paper is focusing our discussion around some of knowledge computing and engineering techniques and approaches and their applications in industrial information systems.

To define the scope of the knowledge computing discipline, a comprehensive analysis has been done of the publications during the last 10 years [3, 5, 6]. It showed that the knowledge computing main disciplines is consist of three main research areas, namely: Document Engineering, Knowledge Engineering, and Reasoning Techniques.

From our analysis of the published literatures about of the knowledge-based systems, we can summarize the general features of knowledge computing approaches in the following.

The knowledge consists of facts, concepts, theories, and procedures A variety of knowledge representation techniques are used including: lists, trees, semantic networks, frames, scripts, production rules and ontology [7]. Lists are used to represent hierarchical knowledge. Hierarchical knowledge can also be represented visually with graphs called trees. Semantic networks use circles called nodes that represent objects or events. The nodes are interconnected with lines called arcs that show relationships. Frames and scripts are two types of schemes dealing with stereotyped knowledge. Frames are used represent facts about objects and events. And details are given in sub-elements called slots. Scripts describe knowledge that is a sequence of events or procedures. Frames and scripts permit a system to infer details of specific common objects and events. Production rules are the most commonly used knowledge representation methods. The rules are two parts statements with a premise and a conclusion and are written in the form of an if-then statement. They also may state a situation and corresponding action [1, 2, 3].

The field of reasoning is very important for the development of knowledge-based systems. The research area in this field covers a variety of topics, e.g.; automated reasoning, case-based reasoning, commonsense reasoning, fuzzy reasoning, geometric reasoning, non-monotonic reasoning, model-based reasoning, probabilistic reasoning, causal reasoning, qualitative reasoning, spatial reasoning and temporal reasoning [1, 7].

Reasoning with Production Rules are easily manipulated by reasoning systems. Forward chaining can be used to produce new facts (hence the term "production" rules), and backward chaining can deduce whether statements are true or not. On the other side, backward chaining is often called goal-directed inference, because a particular consequence or goal clause is evaluated first, and then we go backward through the rules. Unlike forward chaining, which uses rules to produce new information, backward chaining uses rules to answer questions about whether a goal clause is true or not. Backward chaining is more focused than forward chaining, because it only processes rules that are relevant to the question. It is like to how resolution is used in predicate logic. However, it does not use contradiction. It simply traverses the rule base trying to prove that clauses are true in a systematic manner. Backward chaining is used for advisory systems, where users ask questions and get asked leading questions to find an answer. One advantage of backward chaining is that, because the inference is directed, information can be requested from the user when it is needed. Some reasoning systems also provide a trace capability that allows the user to ask the inference engine why it is asking for some piece of information, or why it came to some conclusion.

Unlike Boolean logic, which has only two states, true or false, fuzzy logic deals with truth values which range continuously from 0 to 1. The use of fuzzy logic in reasoning systems impacts not only the inference engine but the knowledge representation itself. Reasoning with fuzzy rule systems is a forward-chaining procedure. The initial numeric data values are fuzzified, that is, turned into fuzzy values using the membership functions. Instead of a match and conflict resolution phase where we select a triggered rule to fire, in fuzzy systems, all rules are evaluated, because all fuzzy rules can be true to some degree (ranging from 0.0 to 1.0). The antecedent clause truth values are combined using fuzzy logic operators (a fuzzy conjunction or and operation takes the minimum value of the two fuzzy clauses). Next, the fuzzy sets specified in the consequent clauses of all rules are combined; using the rule truth values as scaling factors. The result is a single fuzzy set, which is then defuzzified to return a crisp output value.

Reasoning from experience is a natural way of human thinking, one remembers an apparently similar situation, what one has done and what the outcome has been, accordingly one acts in the present situation. The case-based reasoning draws from this paradigm and tries to formalize it for use on the computer [9]. The case-based reasonings the scientific method (or collection of methods) to imitate and enhance, if possible, this human behavior to find useful and applicable old cases and to reuse them either directly or after adaptation. In addition, the success of adaptation has to be verified and cases have to be collected for future use. The case-based reasonings, as a computational intelligence method, assumes a memory model for representing, indexing and organizing past cases and a process model for retrieving and modifying old cases and assimilating new ones [9, 10]. The case-based reasoning has already been applied in a number of application areas, such as customer support and environmental monitoring applications, engineering, and intelligent robotics [9, 10].

The case is a list of features that lead to a particular outcome (e.g. the information on a projects). The complex case is a connected set of subcases that form the problem solving task's structure (e.g. The design of an airplane). Determining the appropriate case features is the main knowledge engineering task in case-based knowledge-based software. This task involves defining the terminology of the domain and gathering representative cases of problem solving by the expert. Representation of cases can be in any of several forms (predicate, frames). The idea of case-based reasonings becoming popular in developing knowledge-based systems because it automates applications that are based on precedent or that contain incomplete causal models. In a rule-based system, an incomplete mode or an environment, which does not take into account all variables, could result in either an answer built on incomplete data or simply no answer at all. Case-based methodology attempt to get around this shortcoming by inputting and analyzing problem data [9, 10].

The fusion of computational intelligence and knowledge computing paradigms with conventional knowledge acquisition from subject matter experts solves many of the problems in recent generation of in industrial knowledge-based information systems. These paradigms play a key role in developing smart and robust tools for engineering tasks. Case-based reasoning methodology seems best suited for the core process of knowledge management because cases can represent knowledge well in terms of knowledge creation, storage and retrieval. Ontological engineering is an effective methodology for accumulating, representing, and management, changing and updating knowledge in intelligent systems.

## References


1. Luger G.F. Artificial Intelligence Structure and Strategies for Complex Problem Saving – Addison Wesley, 2005.
2. Greer J. Proceedings of AI-ED 95, World Conference on Artificial Intelligence in Education, Association for Advancement of Computing in Education (AACE) – 1995.
3. Waterman D. A. A Guide to Expert Systems – Addison-Wisley, 1986.
4. Kane, B. and Rucker, D. W. AI in medicine, AI Expert – Kinnucan, 1998.
5. Salem A.B., Katoua H.S. Web-Based Ontology of Knowledge Engineering, Journal of Communication and Computer – No.9, – 2012, – P. 516-522.
6. Glushko R.J., Mcgrath T. Document Engineering. – MIT Press. – Cambridge, USA, 2005.
7. Michell T.M. Machine Learning – McGRAW-HILL, 1997.
8. Kolonder, J. Case-Based Reasoning. – Morgan Kaufmann, 1993.
9. Salde S. Case-Based Reasoning: A Research Paradigm, AI Magazine, – Vol. 12, – No. 1, – 1991, – P. 42-55.
10. Abdrabou E.A. M., Salem, A.B. Case-Based Reasoning Tools from Shells to Object-Oriented Frameworks. Advanced Studies in Software and Knowledge Engineering, International Book Series "Information Science and Computing", – 2008, P. 37-44.