Expert Systems

Chapter · December 2016

DOI: 10.1002/0471238961.0524160518011305.a01.pub2

Mehmet R. Tolun
Konya Food and Agriculture University

Kasim Oztoprak
Konya Food and Agriculture University

EXPERT SYSTEMS

1. Introduction

Expert systems (ES) are knowledge-based systems that were one of the earlier research fields in Artificial Intelligence (AI) and can be defined as knowledge-intensive software that can perform some tasks normally requiring human expertise. Expert systems are used to solve specific domain problems and each step of reasoning for a specific problem is determined by the human expert professionally. So, they behave as an artificial advisory system for a particular problem domain.

Although AI is used in various commercial applications today, an expert system application is sometimes regarded as “AI” too. After expert systems have moved out of research laboratories during early 1980s, they became more popular and found several application fields such as engineering, chemistry, medicine, industry, and many others.

The construction process of expert systems with specialized domain knowledge is defined as knowledge engineering. Knowledge-based expert systems contain knowledge acquired from periodicals, books, or from domain interviews with human experts. Expert systems are mostly preferred as they produce reasonable solutions for even some ill-structured problems that have no efficient algorithmic solution (1). In addition to classical expert systems, there are hybrid expert systems today using techniques such as artificial neural networks and genetic algorithms.

Broader information for expert systems is given in Section 2. Then, historical development and current applications of expert systems will be mentioned in Sections 3 and 4, respectively.

Q1 2. Expert Systems

2.1. General Concepts.

The first expert systems were built by interviewing an expert and attempting to capture the knowledge, hence the term “expert systems.” An ES is a computer program, which is constructed by utilizing the experience of a domain expert. It performs functions like asking questions and explaining its reasoning. The user interface of this kind of system proceeds with the question answer manner by the end user.

The kernel of an expert system has two main components, namely the knowledge base and the inference engine. The knowledge base contains knowledge about the expert’s domain. It may be represented by simple facts, or by more complex representations like frames. There are also rules that explicitly represent the expert’s skills or knowledge about the domain under consideration. The expert system uses this knowledge by exploiting the second main component, that is the inference engine that has several roles including determining how the system reasons using the IF–THEN rules in the knowledge base. Once the knowledge base is built, the ES can begin making inferences. The most common forms of inferencing are forward and backward chaining. The process of moving forward from known facts to conclusions that follow them is called forward chaining. Alternatively, the process of working backward from a hypothesis to known facts that support it, is called backward chaining.

DOI: 10.1002/0471238961.0524160518011305.a01.pub2

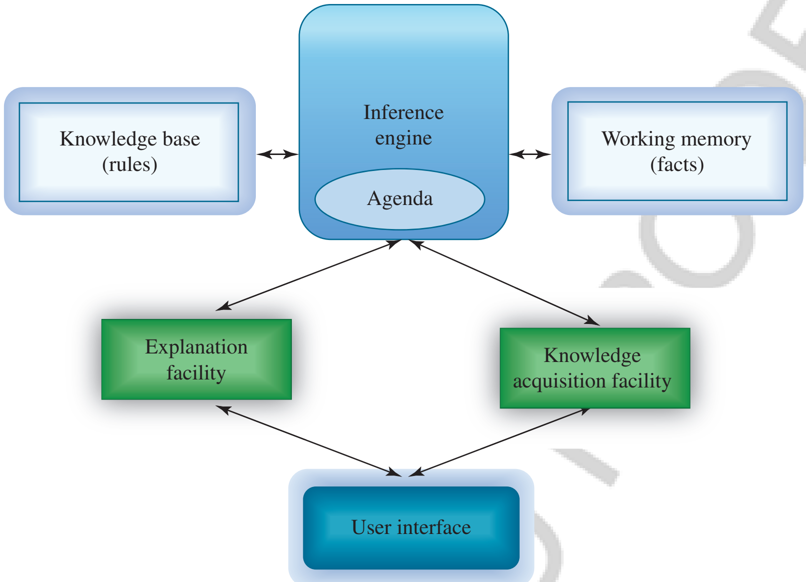

Fig. 1. The general architecture of an expert system (1).

The general architecture of an expert system is presented in Figure 1 and its components are defined as follows (1):

- User interface—the mechanism by which the user and the expert system communicate.
- Explanation facility—explains the reasoning of the system to a user.
- Working memory—a database of facts used by the rules.
- Inference engine—makes inferences by deciding which rules are satisfied by facts or objects, prioritizes the satisfied rules, and executes the rule with the highest priority.
- Agenda—a prioritized list of rules created by the inference engine, whose patterns are satisfied by facts or objects in working memory.
- Knowledge acquisition facility—an automatic way for the user to enter knowledge in the system rather than by having the knowledge engineer explicitly code the knowledge.

Knowledge Elicitation. The first phase of building an ES involves obtaining expert's knowledge. Domain-specific knowledge is extracted from a number of sources using one or more of the following knowledge elicitation techniques:

- Interviewing
- Protocol analysis
- Multidimensional scaling
- Card sorting

Interviewing is used to identify the overall structure of the expert's knowledge domain. Two main approaches are used:

Structured interviews—specific probing questions are asked (scope of the questions are usually identified in a preliminary interview with the expert).

Unstructured interviews—expert is asked to provide all information regarding the domain without specific elaboration of detail. Unstructured interviews can be used as a preliminary interview to the structured approach.

In the protocol analysis, the expert is asked to "think aloud." This is then recorded in some way. There are two forms of protocol analysis:

1. Expert verbalizes every thought and action while solving a problem. This verbalism is recorded and then transferred to script. This method is termed concurrent protocol because the information is obtained at the same time the expert solves the problem.
2. Expert recorded while the problem is being solved. The video is then shown to the expert, who is asked to explain what he was thinking and doing. The method is termed retrospective protocol and is very useful to extract information that is largely tacit and not easy to verbalize.

The two protocol techniques when applied together may provide precise information concerning how a problem is tackled.

Multidimensional scaling is technique to elicit experience and relationships between objects from the view of an expert. Primarily used when there are a number of closely related concepts and no specialized vocabulary to express subtle distinctions and relationships. The technique involves visually representing the psychological similarities between objects or experiences as points on a scatter graph. Objects, which are psychologically dissimilar, are shown far apart; the distance between them can be analyzed to interpret the underlying dimensions as to why these objects have been judged relative to one another.

Finally, card sorting techniques provide means of achieving a more focused or systematic understanding of the classifications and relationships in the expert's domain. It is easy to implement and involves writing the names of objects, experiences, or rules in the expert's domain onto individual cards. Usually implemented either as group separation tasks or group creation tasks. In group separation tasks, the expert is asked to group cards into two, which are then named. The cards are shuffled and then the whole procedure is applied for three, four, and more groupings. Whereas, in the group creation groups are not made smaller but "built-up." The expert is asked to find a pair of cards from the set of cards that are most similar than any other pair.

Rule-Based ES—Knowledge Representation and Inferencing. Rule-based expert systems consist of set of IF–THEN rules as represented in the following (2):

IF the substance x is copper
AND the temperature of x is room temperature
THEN the phase of x is solid.

The IF part is defined as the antecedent that contain fact(s) and the THEN part is defined as the conclusion that contain goal(s). The antecedent part in which facts are executed first is called as data-driven, or forward chaining and the conclusion part in which goal parts of IF–THEN rules are executed first is called as goal-driven or backward chaining. The forward chaining approach is generally preferred in the case of a few and expensive data collection and backward chaining approach is preferred when a specific result is required with respect to large quantity of data (3).

The two main tasks related to reasoning issues addressed by the inference engine are the following:

1. After obtaining the input from the user, the inference engine decides where to start the reasoning process by going through the rules and facts that reside in the static knowledge base.
2. The inference engine resolves conflicts that occur when more than one rule has a consequent matching the current goal. When the system reaches to a point where there are more than a few rules ready to be executed the inference engine decides which rule to examine next. This is called conflict resolution and there are many different strategies to handle this sort of situation.

To summaries, the inference process is carried out in three stages as shown in Figure 2. During the match stage, the contents of working memory are compared to facts and rules contained in the knowledge base. When consistent matches are found, the corresponding rules are placed in a conflict set. Once all the matched rules have been added to the conflict set during a cycle, one of the rules is selected for execution.

Development of an ES. Basically, the expert system development process consists of construction of rules that are derived from problem-solving interviews with a human expert. The knowledge engineer then organizes this knowledge

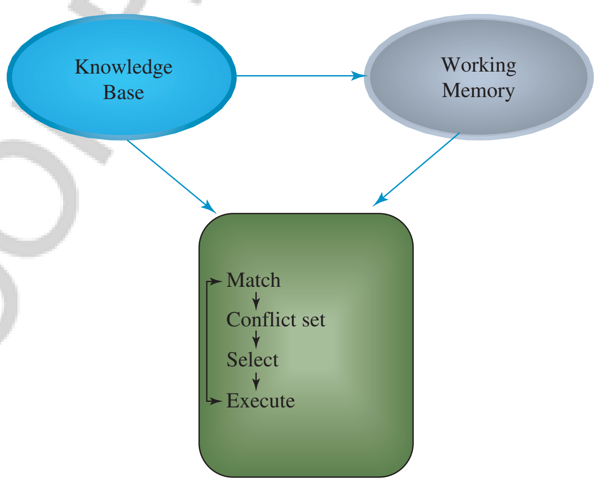

Fig. 2. The inference cycle.

collected from the expert in a form that can be effectively represented in an expert system.

Building an ES is an iterative process involving the creation of a prototype system and then over a number of cycles of testing, repair, and extension, incrementally improving the system so that it eventually performs in a way that is satisfactory and beneficial to the users. The problem identification, conceptualization, formalization, construction, and testing are the stages of ES development process (4).

Once a system has been built and debugged it is a simple matter to extend its coverage of cases at that level of expertise.

Limitations of Expert Systems. The main limitations are the following:

* ES work well only in a narrow domain.
* Knowledge transfer from the domain expert is subject to perceptual and judgmental biases. ES constructed with the help of a single expert can differ in its conclusions if checked independently by another expert in the same area.
* Contradictory information obtained from domain experts can yield faulty conclusions.

2.2. ES Environments (Languages, Tools, And Shells). The efforts of AI scientists directed toward providing aids for inexperienced users and developing for such users a simple, English-like programming languages in 1970s, resulted in rule-based system tools later called expert system shells. These development tools have a user interface where a rule-based ES framework is constructed and run. A shell is a special purpose tool with a user interface and is used to design and implement an application using its built-in features. The very first knowledge engineering tool is EMYCIN that is the domain-independent version of MYCIN, a medical diagnosis ES for diagnosing infectious blood diseases and is a full-fledged shell except the knowledge base (1,5,6). Other tools derived from ES such as PROSPECTOR have led to tools similar to MYCIN. Although conventional programs mainly produce numeric solutions, Expert Systems are designed for symbolic reasoning thus even attempting to solve ill-defined problems. Apart from ES building tools, there are various languages that can be utilized for ES construction such as LISP and PROLOG. The foremost being PROLOG, which is very efficient with symbols, conditions, and statements and can also be used to construct rule bases. However, one should bear in mind that although programming languages are more flexible, they are more difficult to use for developing a prototype of a new system rapidly.

2.3. Application Domains. Designing expert systems for suitable application domains have a great importance on the efficiency of the system. There are several expert system studies in different application domains and these can be classified as Agriculture, Computer Systems, Education, Electronics, Energy-Power, Engineering, Finance-Business, Geology, Law, Human Resources Management, Information Management, Manufacturing, Mathematics, Medical-Health, Military, Space Technology, Telecommunication, and other areas. ES have been built to solve many different types of problems in various areas, but their utilization categories can be grouped as configuration, diagnosis, instruction, interpretation, monitoring, planning, prognosis, remedy, and control. For example, early well-known expert systems are DENDRAL, CRYSALIS, TQMSTUNE, CLONER, MOLGEN, SECS, SPEX, and SYNCHEM2 in chemistry (2,7–9). ACE, EURISKO, SOPHIE, PALLADIO, REDESIGN, TALIB, and CADHELP are in electronics, whereas PUFF, SPE, VM, CADUCEOUS, BLUE BOX, ONCOCIN, GUIDON, MYCIN, are ABEL are earlier examples of medical expert systems (1,10).

3. Historical Development of Expert Systems

3.1. Milestones. During late 1950s several important programming projects were begun including the General Problem Solver developed by Newell and Simon to solve a variety of problems ranging from symbolic integration to puzzles like missionaries & cannibals. In 1960s, forerunners of expert systems have begun by the development of practical programs for real-world problems in different fields. In 1965, the DENDRAL is created as the first knowledge-based expert system. It is designed to discover molecular structures given only information of the chemical constituents of the compound and mass spectra data (11).

In 1970s, the importance of the integrated usage of the production rules, shells, and the knowledge concepts have increased. MYCIN is developed for medical diagnosis for detection of bacterial infections in 1973. It also recommends a list of therapies for the patient. The knowledge base is separated from the inference engine and this feature of MYCIN offers other diagnostic systems to be built rapidly. The initial knowledge base of MYCIN contained about 200 rules. This number gradually increased to more than 600 by the 1980s.

In 1980s, the value of expert systems was well established. A number of successful applications had been completed by then and they proved to be cost effective in most cases using commercial products that have started to emerge. C Language Integrated Production System (CLIPS) expert system tool is used by academia, industry, and the governmental institutions. It is written in C programming language by also employing Rete algorithm for rule matching. CLIPS supports object-oriented, rule-based, and procedural programming, CLIPS can be implemented on computers that use ANSI C compiler, the extended versions of CLIPS (FuzzyCLIPS, COOL, . . .) offer integrated usage of fuzzy logic and object-oriented features. On the other hand, CLIPS does not support backward chaining. In mid-1990s Java version of CLIPS that is called as JESS is produced and has all properties of CLIPS and additionally, it offers powerful rule matching to facts and the backward chaining process is supported (1,3,12).

3.2. Knowledge Acquisition Bottleneck. The process of knowledge acquisition is recognized as one of the major bottlenecks in developing classical rule-based or frame-based expert systems that require conversion of expert knowledge into rules where it may not be straightforward to convert the domain expert's knowledge into a set of logical rules.

Neural-network-based connectionist expert systems and induction-based systems to be described next offered promising approaches to overcome the knowledge acquisition hurdle. These approaches are qualitatively different from the standard expert systems in representation of knowledge and learning.

Connectionist Expert Systems Based on Artificial Neural Networks. A neural network can, in fact, serve as the knowledge base for an expert system that does classification tasks. For a neural network expert system, knowledge representation consists of a network, connection weights, and semantic interpretations attached to cells and activations. The main advantage of this approach is that the underlying learning algorithm, such as Backpropagation, can take training examples and generate expert systems automatically. This procedure is illustrated in Figure 3. Typical applications include management and administration (cost estimation, scheduling), industrial (process control, manufacturing quality control, fault diagnosis), medical (medical diagnosis in specialized domains, bacteria identification), banking (credit and loan decisions), and other fields (forecasting). If training data is available then any domain becomes a candidate for a connectionist expert system.

The neuron model developed by McCulloch and Pitts in 1943 can be regarded as the starting point for the connectionist expert systems. Also, a learning process of neurons is defined by Hebb in 1949, and the efficiency of this Hebbian neuron learning model is determined by the emulation and transferring each impulse of one neuron to another neuron successfully and this process is defined as firing rules between each neuron.

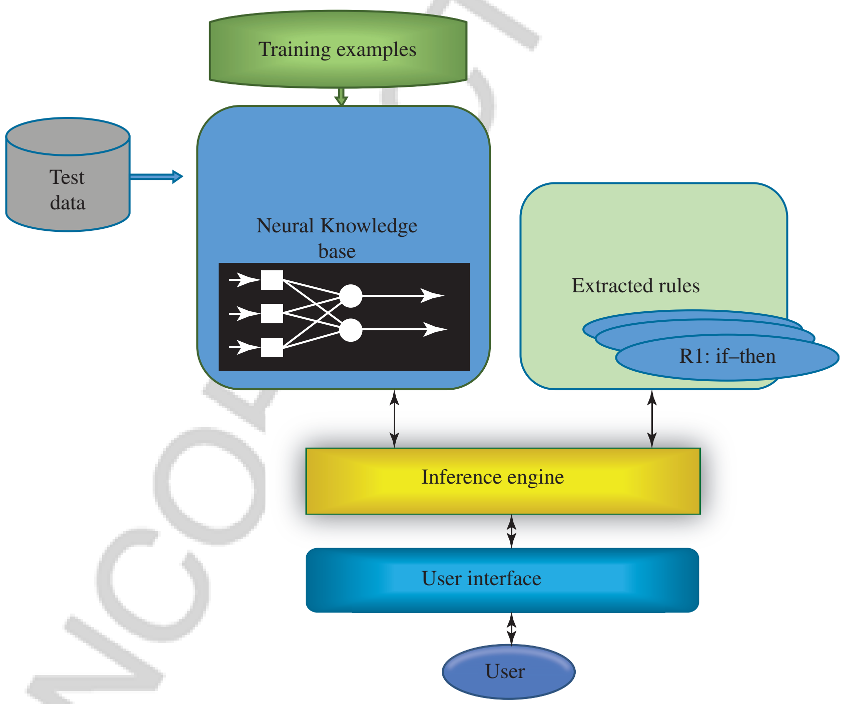

Fig. 3. Extracting rules from neural networks.

Inductive learning is used in the neural network expert systems or connectionist expert systems, and these types of expert systems are advantageous when there is much empirical data and also it is used to prevent knowledge acquisition bottleneck (13–15).

The expert system rule application to define training and test patterns is represented in the following medical expert system example. The knowledge base of this expert system consist of various IF–THEN rules that are related with diagnosing illnesses to achieve the more appropriate treatment and this system can be implemented as a three layered-neural network as in the following example (13).

The neural network consists of cells that correspond to symptoms in the input layer, the diseases are presented in the intermediate or hidden layer and neural cells (nodes) for the treatments defined in the output layer. Training patterns consist of 0's (lack of knowledge about presence or absence of the disease), 1's (presence of the disease), and −1's (absence of the disease). A sample rule from Hepar, a medical expert system for the diagnosis of liver and biliary tract diseases is (16)

IF

duration (x, chronic) **and**
disorder (x, hepatocellular) **and**
age (x, <25) **and**
lab-results (x, Kayser–Fleischer rings)
THEN

diagnostic (x, Wilson's disease)

The general topology for this connectionist ES consists of a two-layer neural network and is shown in Figure 4. In this structure, the diagnosis of the related disease is activated only certain symptomatic situations are identified as true ("1"). At the same time, this system is trained and tested by one of the supervised learning algorithm such as backpropagation algorithm. However, we should note that not all expert system problems are suitable for a neural network approach. The most suitable problems are those that seek to classify inputs into a small number of groups.

Induction-Based Systems. The major bottleneck of building expert systems lie in knowledge elicitation from domain experts. There are several difficulties with acquiring knowledge from an expert.

1. Knowledge mismatch, the difference between the way expert's own knowledge is structured and the way it is represented in the program.
2. Inability of humans to express knowledge they possess and the inherent nature of knowledge (subconscious, approximate, incomplete, inconsistent, etc).
3. Problem of verification and validation.

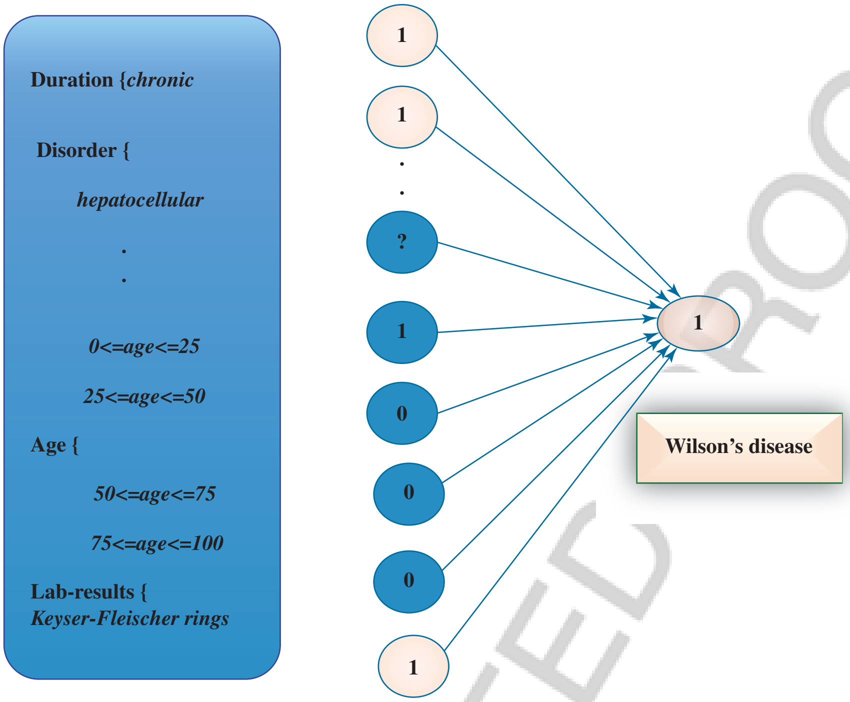

Fig. 4. Diagnosing the Wilson's disease representative example (16).

Major benefits of using automated methods, such as induction-based systems, are

1. It might be more competent than humans for acquiring or fine-tuning certain kinds of knowledge.
2. It might significantly reduce the high cost in human resources involved in construing the system.

One of the approaches to relieve the expert of much of the burden of authoring rules directly is inductive machine learning that is often used by forming a decision tree from a set of training examples. Other methods include rule induction where classification rules are generated by a learning algorithm over training examples (17). An example is expressed a vector of values pertaining to attributes of the decision, together with the expert's classification.

4. Current Applications and the Future of Expert Systems

4.1. Current Applications. Current applications of expert systems and AI are mentioned in the bibliographic studies for expert systems (10,18–21). Expert system implementation methods can be classified as classical expert systems, neural expert systems, fuzzy expert systems, rough set-based expert systems, multi-criteria decision making expert systems, neuro-fuzzy expert systems, web-based expert systems, and multi-agent expert systems.

Today, expert system applications are frequently developed on commercial shells (22)[1]. EXSYS Corvid (for Windows)/EXSYS Corvid Core (the same tool for Mac OS X) is one of the popular shells. In addition, there a quite number of free general purpose and specific ES building tools available[2]. The recent ES applications are mostly web-based or Internet-based (23–27). Web-based systems for delivering online advice have been available since mid-1990s. Unfortunately HTML, a common language for information distribution on Web pages, cannot run expert systems. Therefore, many techniques for implementing Web-based inference engines have been created. For instance, the Configurate and Sizing Tool/SAP (CAST) were developed by Hewlett Packard to aid in configuring SAP business information warehouse implementations. The Coating Alternatives Guide and Solvent Alternatives Guide, both developed by the Pollution Prevention Program at the Research Triangle Institute, have the goal of reducing industrial pollution. Both systems are rule-based expert systems, developed in a custom shell written in ColdFusion, a Web application server framework (23).

A review of Internet-based expert systems and a case study that is “the Reptile Identification Helper” are presented in Reference (23). A meta-model development process for web-based expert systems and Landfill Operation Management Advisor (LOMA) web-based expert system that is based on this meta-model structure is given in Reference (24). A web-based expert system for flood water level prediction in an area is provided in Reference (25). Also, there are some interesting applications related with designing web-based semantic expert systems (26,27).

Another popular Internet-based expert system is found at http://www.MyMajors.com, which provides advice to high school students or college freshmen who are seeking assistance in selecting a potential major by emulating a professional academic advisor (28).

5. The Future

Expert systems have now reached the maturity stage in their development. The early research has established the viability of this approach. Many commercial systems have been developed since then to show their use in real environments. These days we find them either as hybrid expert systems or embedded on web pages in the Internet (29–33).

The basic limitation currently, is to build expert systems with heuristic and empirical knowledge rather than deep knowledge, which include models of the functional and causal relations that underlie a problem. In the future, more systems might be developed using functional and causal models using a variety of representations.

Using multiple sources of knowledge (i.e., domain experts) in a cooperative manner is still a difficult problem waiting to be tackled.

[1] Exsys Corvid, http://www.exsys.com/exsyscorvid.html.
[2] SourceForge, https://sourceforge.net/directory/os:windows/?q=expert%20system

Knowledge-based expert systems will continue to increase individual and social potential by preserving know-how, distributing knowledge more effectively, and improving performance of tasks that require expertise.

BIBLIOGRAPHY

"Expert Systems" in *ECT* 4th ed., Vol. 9, pp. 1082–1112, by T. S. Ramesh; published online: Dec. 4, 2000.

CITED PUBLICATIONS

1. J. Giarratano and G. Riley, *Expert Systems-Principles and Programming*, 4th ed., Course Technology, 2004.
2. J. Klaessens and G. Kateman, *Fresenius J. Anal. Chem.* **326**, 203–213 (1987).
3. A. Abraham, *Rule-Based Expert Systems, Handbook of Measuring System Design*, John Wiley & Sons, 2005.
4. K. P. Tripathi, *Artif. Intell. Tech.—Nov. Approaches Pr. Appl.* **4**, 21–25 (2011).
5. Q2 E. C. Ogu, Y. A. Adekunle *Int. J. Sci. Res.* **2** (4), 554–559 (2013).
6. H. H. Owaided and M. M. Qasem, Developing Rule-Case-Based Shell Expert System, in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, March 17–19, 2010, vol. **1**.
7. B. A. Hohne and T. H. Pierce, *Expert System Applications in Chemistry*, ACS Symposium Series; American Chemical Society, Washington, DC, 1989.
8. G. S. T. Streitberg, P. T. Bwititi, L. Angel, A. Kenneth, and K. A. Sikaris, *Automation and Expert Systems in a Core Clinical Chemistry Laboratory*, The Association for Laboratory Automation, 2009.
9. Q. Zhu, Application Of Expert Systems In Analytical And Environmental Chemistry, 1996, Ph.D. Thesis, Department of Chemistry, The University of Western Ontario, *Digitized Theses*. Paper 2632, 1996.
10. S. Sahin, M. R. Tolun, and J. G. Hall, *Expert Syst*. 381–384 (2013).
11. E. A. Feigenbaum and B. G. Buchanan, *Artif. Intell.* **59**, 233–240 (1993).
12. S. Kumar and R. Prasad, *Int. J. Innov. Res. Dev.* **4** (3), 128–133 (2015).
13. F. N. Alpaslan and M. R. Tolun, Connectionist Expert Systems: An Approach to Solve Knowledge Acquisition Bottleneck in Knowledge Engineering, *Proceedings of Artificial Neural Networks and Artificial Life Symposium, Dec. 15, 1994, METU, Ankara, Turkey*.
14. M. C. Hemmer, *Expert Systems in Chemistry Research*, CRC Press, 2007.
15. S. I. Gallant, *Commun. ACM* **31** (2), 152–169 (1988).
16. P. J. F. Lucas, The representation of medical reasoning models in resolution-based theorem provers. *Artif. Intell. Med.* **5** (5), 395–414 (1993).
17. C. T. Leondes, ed., *Expert Systems, The Technology of Knowledge Management and Decision Making for the 21st Century*, Academic Press, 2002.
18. P. Vizureanu, *Expert Systems*, Intech, 2010.
19. S. Sahin, M. R. Tolun, and R. Hassanpour, *Expert Syst. Appl.* **39**, 4609–4617 (2012).
20. A. Bahrammirzaee, *Neural Comput. Appl.* **19**, 1165–1195 (2010).
21. T. B. Cross, *Knowledge Engineering 2016 The Uses of Artificial Intelligence in Business*, TECHtionary Corporation, 2016.
22. E. Awad, *Building Knowledge Automation Expert Systems with Exsys Corvid*. Exsys Inc. 2003.
23. R. Grove, *Expert Syst*. **17** (3), 129–135 (2000).

24. I. M. Dokas and A. Alapetite, *A Development Process Meta-Model for Web Based Expert Systems: The Web Engineering Point of View*, Risø National Laboratory, Denmark, 2006.
25. R. Islam and K. Andersson, *Mohammad Shahadat Hossain, A Web Based Belief Rule Based Expert System to Predict Flood*, WAS2015, Brussels, Belgium, Dec. 11–13, 2015.
26. O. Verhodubs and J. Grundspenkis, Towards the Semantic Web Expert System. *Appl. Comput. Syst.* **44** (1), 116–123 (2012).
Q3
27. M. Nofal and K. M. Fouad, *Int. J. Comput. Sci. Issues* 11, **1** (1), 103–110 (2014).
28. F. H. Grupe, *The Internet and Higher Education: A Quarterly Review of Innovations in Post-Secondary Education*, Jai Press, Greenwich, Conn., **5** (4), 2002, pp. 333–345
29. S. Wang and R. A. Noe, *Hum. Resource Manage. Rev.* **20**, 115–131 (2010).
30. J. K. Nurminen, O. Karonen, and K. Hätönen, *Expert Syst. Appl.* **24**, 199–211 (2003).
31. J. T. Ball and R. F. Moody, The Future of Expert System Development Tools? ORION International Technologies, Inc., The Fifteenth Annual Ideas in Science & Electronic Exposition and Symposium'93, 1993.
32. C. F. Tan and co-workers, *ARPN J. Eng. Appl. Sci.* **11** (4), 2448–2453 (2016).
33. C. Angeli, *Appl. Res.* **1**, 50–73 (2010).

Mehmet R. Tolun
Aksaray University, Aksaray, Turkey

Seda Sahin
Baskent University, Ankara, Turkey

Kasim Oztoprak
KTO Karatay University, Konya, Turkey

Keywords/Abstract

Dear Author,

Keywords and abstracts will not be included in the print version of your article but only in the online version. The abstract and keywords are included in the print version only for your review.

Abstract:

Expert systems have emerged around mid-1970s under the umbrella of Artificial Intelligence and as soon as convincing success was attained, the field was transformed into an established branch of computer science. The potential of expert systems that emulate human knowledge and skill has also encouraged the development of many applications in various areas.

Expert systems contain specialized knowledge elicited from a domain expert. Various expert system building tools or shells exist to greatly facilitate and speed up the development of expert systems. Some of the new generation tools also allow an expert system to be delivered over the Internet.

This article introduces expert system and knowledge engineering concepts and discusses issues related to expert system design and development in various areas.

Keywords: expert system shells; induction-based expert systems; knowledge-based systems; knowledge engineering; neural network expert systems.

Author Query

1. Please check order of the section headings for correctness.
2. Please check Author name of Reference No. (5) for correctness.
3. Please check the changes made.