

The Impact of Intelligent Interfaces on the Performance of Knowledge Base Systems

Abid Thyab Al Ajeeli

Higher Education Committee,

C. O. R., Iraq.

Type of Review : Peer Reviewed.

DOI: <http://dx.doi.org/10.21013/jte.v4.n2.p1>

How to cite this paper:

Ajeeli, A. (2016). The Impact of Intelligent Interfaces on the Performance of Knowledge Base Systems. *IRA-International Journal of Technology & Engineering* (ISSN 2455-4480), 4(2), 84-102. doi:<http://dx.doi.org/10.21013/jte.v4.n2.p1>

© Institute of Research Advances



This work is licensed under a [Creative Commons Attribution-Non Commercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) subject to proper citation to the publication source of the work.

Disclaimer: The scholarly papers as reviewed and published by the Institute of Research Advances (IRA) are the views and opinions of their respective authors and are not the views or opinions of the IRA. The IRA disclaims of any harm or loss caused due to the published content to any party.

ABSTRACT

Although Knowledge is recognized as a strategic force in organizations, knowledge creation and management is not simply the capture and storage of items of information. It requires the storage and processing of associations through which meaning can be derived from the information. Association can be represented in explicit and observable forms in a knowledge base. The paper discusses issues relating to design a better quality interactive interface system for human to have a dialog with the knowledge management systems. At the same time, the paper investigates the ease of evaluation and implementation of a knowledge management system. It performs a major role in providing users with capabilities of dealing with underlying systems. Designing a good interface style using knowledge bases can have a profound effect on the nature of the dialog.

Interfacing a knowledge base oriented system can be seen as a dialog between the knowledge base and the user. It plays a major role in providing users with capabilities that deal with underlying systems. Designing a good interface style can have a profound effect on the nature of the dialog. Design of a user interface involves determining approaches in which users interact with the knowledge-based system. The design process, can be complex and multifaceted, begins by identifying system users, through classification of them to understanding their characteristics. The study investigates the improvement of performance by performing an actual study of many experimental systems sufficient to provide judgments for taking the right decision. Results confirm that a good interface has a great impact on the performance of knowledge management systems.

Keywords: Heuristic, Knowledge Base, Knowledge sharing, Knowledge exchange, Knowledge management, Flat Knowledge Base, Rules, Intelligent, Decomposition

1. Introduction

Knowledge adaptation is gaining widespread attention as a strategic tool for the competitiveness of organizations both in the management literature and in popular press. A **knowledge base (KB)** is a technology used to store complex structured and unstructured information used by a computer system. The initial use of the term was in connection with expert systems which were the first knowledge-based systems. However, knowledge creation and management is not simply the capture and storage of information. It also requires the storage and processing of associations (rules) through which meaning can be derived from the items of information and can be stored in explicit and observable form.

The original use of the term knowledge-base was to describe one of the two sub-systems of a knowledge-based system. A knowledge-based system consists of a knowledge-base that represents facts about the world and an inference engine that can reason about those facts and use rules and other forms of logic to deduce new facts or highlight inconsistencies [1].

Knowledge management (KM) is the process of capturing, developing, sharing, and effectively using organizational knowledge. It refers to a multi-disciplinary approach to achieving organizational objectives by making the best use of knowledge [2, 3]. In today's world data are so numerous that technology is needed to cope with this knowledge. KM which interacts with knowledge based systems contains processes and/or tools that involve sorting the collected items of information and select those that are relevant. KM provides critical insights that help organizations make right decisions.

Knowledge management is a key approach to managing business environments and allowing employees to obtain relevant insights and understanding appropriate to their work and making available increased knowledge content in the development and provision of products and services. KM can explicitly be defined as a systematic process or a discipline that promotes an integrated approach to identifying, capturing, evaluating, retrieving, sharing all of enterprises' information assets. It will present information in a way that improves an employees' comprehension in areas of interest. These assets may include databases, documents, policies, procedures, and previously un-captured expertise and experience in individual workers

KM and knowledge base can also benefit from each other. The integration of KM with knowledge base enables organizations achieve wider benefits. Integration of KM with knowledge base will not only help to promote and enhance knowledge for better decision making, but also improve organizations performance. Therefore it is imperative for organizations to have both KM and knowledge base as an integrated system to get full value from both. This work shows the importance of KM and knowledge base Integration through a series of models and shows how the performance be improved [4].

Interfacing a knowledge base oriented system can be seen as a dialog between the knowledge base and the user. It plays a major role in providing users with capabilities that deal with underlying systems. Designing a good interface style can have a profound effect on the nature of the dialog. Design of a user interface involves determining approaches in which users interact with the knowledge-based system. The design process, can be complex and multifaceted, begins by identifying system users, through classification of them to understanding their characteristics.

Typically, an intelligent interface is required such that it should employ some kind of intelligent technique. The concept of intelligent technique may vary over time, but the following list is a fairly complete list of the kinds of techniques that today are being employed in intelligent interfaces [5, 6]:

User adaptively: Techniques that allow the user - system interaction to be adapted to different users and different usage situations.

User modeling: Techniques that allow a system to maintain knowledge about a user.

Natural language technology: Techniques that allow a system to interpret or generate natural language utterances, in text or in speech.

Dialogue modeling: Techniques that allow a system to maintain a natural language dialogue with a user, possible in combination with other interaction means (multimodal dialogue),

Explanation generation: Techniques that allow a system to explain results to users.

Providing the above list of technologies does not capture the essential feature of the intelligent interface research area. An intelligent interface must utilize technology to make an improvement. The resulting interface should be better than any other solution. It is not just different and technically more advanced. There are a number of common interface styles including [5]:

- Command line interface
- Menus
- Natural language
- Question/answer and query dialog
- Voice output to guide users or answer queries
- Form-fills and traditional end user applications
- Source documents
- WIMP
- Point and click
- Three-dimensional interfaces

WIMP interface is the most common and complex. The most promising one is mixing WIMP with natural language and/or question/answer and query dialog, where **WIMP** stands for windows, icons, menus, pointer.

Every knowledge-based system has its own knowledge formalism depending on the problems to be solved, goal to be achieved, and what a proposed solution to be accomplished. In other words, knowledge designed for one system differs for knowledge types defined for other systems. One of the solutions to overcoming this problem is providing a unified model or a knowledge management interface that accepts different types of knowledge, which guarantees automatic interaction between the knowledge-based systems. To facilitate the understanding of the work in this paper, an interaction is defined as knowledge sharing, integration, and transfer from one system to another. The knowledge

management interaction helps accelerating the establishment of a new knowledge-based system as it does not need knowledge initialization [7].

The paper is organized as follows. The related works on knowledge management systems are investigated in Section 2. Knowledge-based Systems development is outlined in Section 3. Empirical analysis of knowledge base design is outlined and thoroughly studied in Section 4. In Section 5, the impact of knowledge base development on knowledge management is shown in details. Findings and conclusions are summarized in Section 6.

2. Previous Work

Knowledge Management (KM) is a relatively new form of collaborative computing. KM goals are to capture, store, maintain, and deliver useful knowledge in a meaningful form at anytime to anyone needs it and anywhere within an organization. The concept of KM has its roots in the concepts of organizational learning and organizational memory. Knowledge is usually transformed and transferred from individual to individual through the collaboration and communication of ideas, teaching processes, learning, and innovation [8, 9]. Siemens AG central board [10] created an organizational unit to be responsible for deploying KM to the entire branches of the company worldwide. Siemens realized that IT was only the tool that empowers KM. It also found that intelligent interfaces play a major role in participating in the exchange of ideas and experiences [11].

Turban et al. [12] defined KM as a process that helps organizations identify, select, organize, disseminate, and transfer essential items of information and expertise that reside within the organization in an unstructured manner. Turban et al. [12] claimed that the structure of knowledge enables effective and efficient problem solving, dynamic learning, strategic planning, and better decision-making. Using information technologies to make knowledge management available become known as knowledge management systems [13].

Knowledge management is a set of processes of elicitation, transformation of intellectual assets, and diffusion throughout an organization so that it can be shared and reused [14]. It involves a strategy commitment to improving the organization's effectiveness and provide opportunity enhancement. Knowledge management has three main functions as in figure 1 [15].

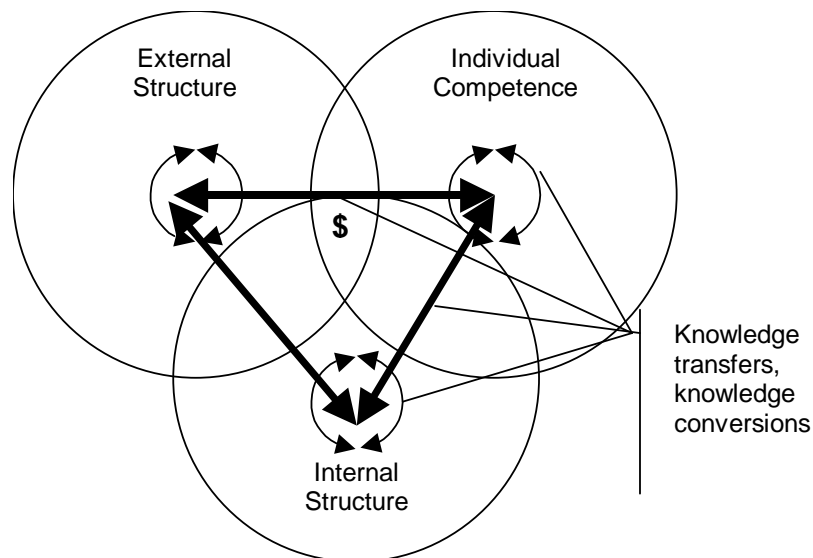


Figure 1: Knowledge transfer

These three main functions are:

1. External structure: which captures knowledge in an external repository and organize it to discover similar knowledge? It is supported by imaging systems, databases, workflow, ..., and document management systems using clustering techniques.
2. Internal structure: which identifies knowledge, usually explicit, relevant to a particular user's need? It involves mapping a particular problem, situation, and/or point of interest against the body of knowledge captured through external structure.
3. Individual competence: Which deals with knowledge management concept that is the most difficult to automate because it relies on individual capabilities to recognize where and how knowledge can be used.

Alavi and Leidner [16] defined KMS as the use of modern information technologies (the internet, intranets, extranets, software, tools, filters, agents, data warehouses) to systematize, enhance, and expedite intra- and inter-firm knowledge management. In other words, Knowledge management systems refer to any kind of IT system that stores and retrieves knowledge, improves collaboration, locates knowledge sources, mines repositories for hidden knowledge, captures and uses knowledge, or in some other way enhances the KM process. In figure 2 a detailed list of elements of knowledge management system [47] has been shown.

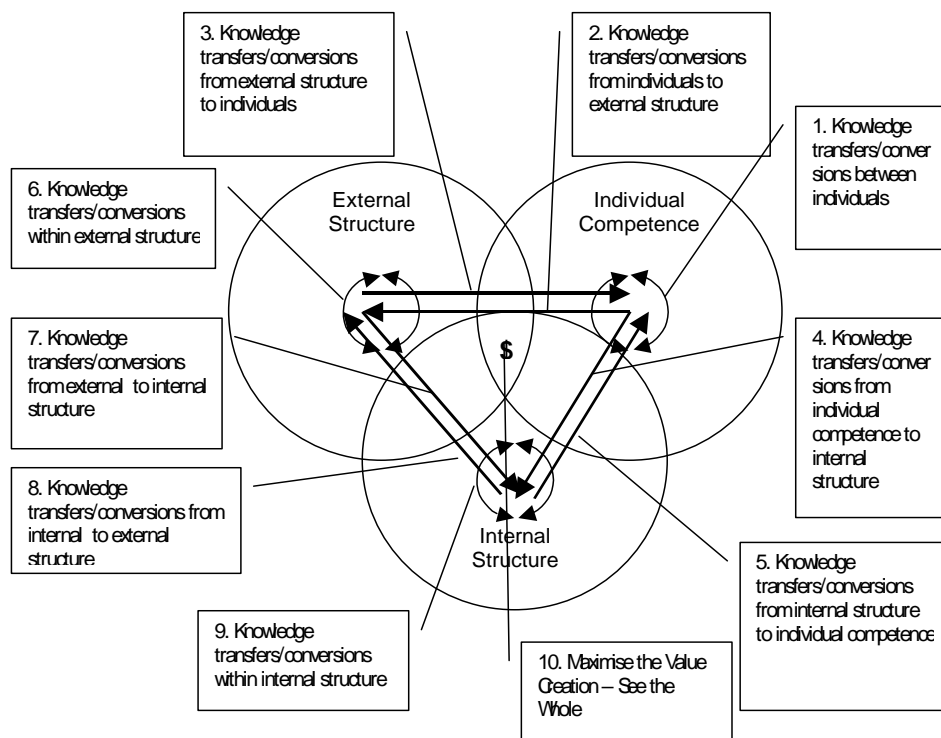


Figure 2: Details of knowledge functions

Cingular wireless is a major mobile communications provider based in Atlanta, Georgia, claimed that KMS is one of the key factors of organization systems success [17, 18]

Encouraging employees to use a knowledge management system for contributing and/or seeking knowledge can be difficult for a number of reasons including [19]

- Lack of skills in knowledge management techniques
- Lack of appropriate technology
- Available technology does not match the organization's culture.

For the above reasons, we need to incorporate technologies to enable advancing functionality of modern knowledge management systems and also to form a base for future innovations in KM field [20, 21]. One of these technologies that support knowledge management is knowledge-based systems.

The rest of the article is organized as follows. First, the knowledge-based systems development is outlined. Second, an empirical analysis of end user developed applications is introduced, together with metrics for knowledge-based system quality. This is followed by a more detailed analysis of the problems encountered in end user development. The purpose of this section is to illustrate to the reader which aspects of the development process create difficulties for the inexperienced developer. Thereafter, the issue of knowledge explication is discussed. The last section extrapolates the findings to determine the limits of end user developed knowledge-based systems and discusses alternative approaches to capture the valuable knowledge of end users.

Trend of globalization has induced a fierce competitiveness among business firms within domestic and/or international markets. Using technology by firms will empower their work context by utilizing its resources. Knowledge management has emerged as the latest techno-management trend for improving the work process and creating value for business firm operations. Knowledge management offers various techno-managerial implications to business firm for strategic development. However, there are scarce evidences on business intelligence, strategic management decision support related to business organization adopting these offerings. Major objective of knowledge management is to extract the information and find the hidden knowledge from all sources of data.

Knowledge management offers to make decision for enhancement of any firms' goal. The broad overview of research articulates an understanding of government based firms about the adoption of Knowledge management based solutions and its challenges. An interface plays a key role in Knowledge Management based systems for business firms and its implication lies in the implementation algorithms for exploring the huge amount of data, which determines the pure knowledge. Majority of the firm data remains in either unstructured form such as raw form of data (i.e. internal or external documents) or with its employees in the form of experience [22]. Knowledge management process deals with extraction of both tacit and explicit knowledge of firm for improving its performance. Interface on the other hand gained its importance with constant enhancement in technologies and tools for extracting the hidden knowledge and patterns. Interface and Knowledge Management are complimentary to each other for extracting and managing the knowledge.

Luo al et. [23] has stated that research on knowledge-based system has focused on the application level, which is how to establish a knowledge-based system and utilize it for a specific purpose. This knowledge-based system will answer a specific problem in a specific domain of knowledge. In the other hand, document classification, where a knowledge-based system is assigned a task to retrieve documents from a specific source, conduct information extraction from the document to obtain the meta data, and create a classification using the meta data as outlined by **Wan** al et. and **Wang** al et. [24, 25]. **Wang** al et. and **Schlenoff** et al. [25, 26] have explained real time information, where a knowledge-based system acquires knowledge from a real time source with spatial information. Knowledge management utilization, is outlined by **Eri** al et.[27], where a knowledge-based system is built and managed for special purposes, while **Huang** al et [28] discussed multilingual information, where a knowledge-based system acquires knowledge from multilingual source, including tacit knowledge. In a recent study by **Maes** [29] addressed the issue of traditional end user applications errors and their potential impact on organizations. **Maes** [29] reports a picture of unacceptably high error rates and significant bottom-line impact related to these errors.

The research in this paper focuses on how an interface and KM integration affect the business firm while discussing its implementation challenges. It tries to analyze the correlation between Knowledge Management and Interface by exploring a road map for data framework for Knowledge Management focusing on based firms. Current situation of knowledge management strategic decision making and

role of knowledge must need to be addressed before proposing any framework for business firms. The work discusses in an extensive details the empirical analysis of knowledge base development, which aims to describe the basics of Knowledge Management based systems and integrating Business interface with Knowledge Management.

3 Knowledge-based Systems Development

A knowledge-based system is a computer program that can reason and use a knowledge base to find solutions to complex problems. The term is broad and is used to refer to many different kinds of systems. The most common theme that unites all knowledge based systems is an attempt to represent knowledge explicitly via tools such as ontology and rules rather than implicitly via code the way a conventional computer program does. The knowledge base represents facts about the world, often in some form of subsumption ontology while inference engine for knowledge base system represents logical assertions and conditions about the world, usually represented via IF-THEN rules.

Knowledge-based system development is more complex compared with quantitative applications development, which can result in even higher error rates, namely [30]:

- Knowledge-based systems are based on qualitative (logic based) reasoning using non-numeric symbols whose semantics can be defined by user rules. Traditional end user applications are largely quantitative with well-established semantics.
- Knowledge-based system developers usually have little training in the “language” of knowledge-based systems, which might be a derivative of first order logic, or a similar logic oriented language. In contrast, traditional end user application developers can rely on many years of training and experience in the common “language” of traditional end user applications.
- Most knowledge-based system shells separate development from use. That is, the knowledge base will be developed in an editor or other development environment (with limited error checking). After completion of the knowledge base, it will then be executed. This separation of development and use makes verification and correction of errors significantly more tedious compared with traditional end user applications.
- Rule-based knowledge-based systems frequently adopt a backward reasoning approach whose logic is not obvious to inexperienced developers, or to developers with experience in traditional programming languages.

The above characteristics add extra complexity that affects knowledge extraction and formalization (extraction and formalization of “rules”), representation (especially in a backward reasoning environment), validation, and learning/usage (logic with user definable semantics versus algebra). Consequently, higher error levels are expected compared to typical end user applications such as traditional end user applications. Special attention needs to be drawn to key sources of errors.

Given the apparent difficulty of end user knowledge-based system development on one hand and the increased demand for end user developed knowledge-based systems on the other hand, the decision should be made to investigate the feasibility of knowledge base creation. Two questions should be addressed.

1. *are end users able to develop knowledge-based systems that are structurally of sufficient quality to represent the knowledge formalization and representation?, and*
2. *are knowledge bases capable of expressing their logic reasoning sufficiently to create valuable ones, knowledge acquisition?*

The first question is investigated by assessing the quality and size of end user developed knowledge bases. The purpose is to determine the limits for the size and reliability of end user created solutions

and thus to assess how far end user knowledge-based system development can go. The second question is addressed through the review of research in the area of tacit and implicit reasoning. Its purpose is to find evidence of whether end-user developers can develop knowledge-based systems that convey useful business knowledge regardless of design quality.

Facts arise when objects are described by attributes, for insurance accounts receivable increases. A fact is one type of a conceptual relationship, in addition to rules. Accounts receivable continues a payment collection problem. This utterance can be interpreted in figure (3) as follows [31]:

```
IF    Account receivable increases
AND  Sales turnover remains constant
THEN this indicates a payment collection problem
```

Figure 3: If analysis structure

A number of examples regarding rules in a knowledge base system [46] outlined in figure (4).

```
IF    Direct-liquidity = satisfactory
AND  General-liquidity = not-satisfactory
THEN liquidity = medium

IF    Financial-status = very-satisfactory
AND  Qualitative-evaluation = satisfactory
OR   Qualitative-evaluation = very-satisfactory
OR   Qualitative-evaluation = perfect
THEN Expert-system-evaluation = very-satisfactory
```

Figure 4: rules in a knowledge base

When work experience is ‘not satisfactory’ (negative work experience) then the expert assumes that the organization of the firm cannot be perfect. This meta-rule initially checks the set of possible (legal) value of the criterion organization, excluding the value ‘perfect’ as shown in figure (5).

```
IF kbentry (L:legalvals(organization)=[not-satisfactory, medium, satisfactory, very-
satisfactory, perfect, unknown])
AND work-exp = not-satisfactory
AND do(add L:legalvals(organization)=[not- satisfactory, medium, satisfactory, very-
satisfactory, unknown])
THEN set legalvals(organization).
```

Figure 5: Meta Rules

Knowledge management uses Meta data as described above, as well as, decision support systems (DSS). Decision support systems are a class of computerized information system that supports decision-making activities. Decision support systems are designed artifacts that have specific functionality. Five more specific decision support system types include communications-driven, data-driven, document-driven, knowledge-driven, and model-driven systems. Communications technologies are central to communications-driven DSS for supporting decision-making. Data-driven DSS provide access to large data stores and analytics to create information. Document-driven DSS use documents to provide information for decision making [32].

Knowledge-driven DSS are sometimes generically called *expert systems* or *recommender systems*. Model-driven DSS use quantitative models for functionality and have been called *model-oriented DSS* and *computationally oriented DSS*. The term *knowledge management system (KMS)* encompasses both document and knowledge-driven DSS. Holsapple and Whinston [48] proposed a slightly different classification identifying text-oriented DSS, database-oriented DSS, spreadsheet-oriented DSS, solver-oriented DSS, rule-oriented DSS, and compound DSS [32].

4. Empirical Analysis of Knowledge Base Design

Empirical studies are essential issue in evaluating and measuring the performance of a knowledge base systems or decision Support systems. Empirical analysis helps in the assessment of knowledge base design and products performance. Analysis covers a number of issues related to knowledge sharing among members. For example sharing knowledge base reports, methodologies, documents, manuals, models, experience, know-how, know-where, know-whom, and how frequently information technology has been used such as bulletin board, email, e-document management systems, knowledge repository[49].

The empirical analysis for the purpose of this study is used to assess a number of knowledge based systems. The end users developers, in our case, are final year students at University of Bahrain. These individuals selected a lower-end development environment, based on simplicity and cost, and chose application areas according to their own knowledge and interests. To study this form of development and the problems associated with it, 30 knowledge-based system development projects were analyzed.

The project counted for 40% of the participants' course grade, thus offering a strong incentive to do well. Participants also had the ability to do well, as they received clearly formulated evaluation guidelines, including a diagnostic sheet. Participants also submitted milestone reports and received feedback on their milestone reports with respect to fulfillment of the evaluation criteria.

4.1 Performance Measurement

Measuring software performance has long been recognized as a complex issue. In the software engineering literature performance is broadly defined as conformance to specifications and is operationalized through attributes such as those identified by McCall et al [33]. These guidelines are useful for professional software development, but have less formality for end user development, where specifications may not formally be outlined or documented.

This study therefore adopted an approach to measure three criteria which are applicable to a variety of development projects, whether systems professional or end user. These are [50, 51, 54]:

- *Completeness* is related to the content of the software to be created.
- *Decomposition* reflects good design thinking, but also strongly impacts maintainability of the knowledge base.
- *Probabilistic reasoning* reflects a key aspect of knowledge-based systems, namely their heuristic nature. As is the practice in measurement of software quality criteria (at least two of them) are measured through different metrics.

The **first** criterion was *completeness* which is almost impossible for end user developed applications to assert whether they are “complete”, or what their level of completeness is. Two measures were chosen:

1. Absolute size of the knowledge base measured by the number of logic rules. The size can be considered a surrogate for richness of the knowledge base, as it determines the number of conditions the knowledge base can handle.
2. Presence of “holes” in the knowledge base, which is a determination of knowledge base incompleteness. A knowledge base hole exists, when there are conditions for which the knowledge base has no response, even though they belong to same domain.

Logical Checks for Completeness and Incompleteness of a knowledge base is the result of missing rules. This situation exists in which a particular result is required, but no rule can succeed in that situation to produce the desired result. Missing rules can be detected logically if it is possible to enumerate all circumstances in which a given decision should be made or a given action should be taken [34]. For example, the knowledge base may have investment for the inflation market fluctuation but has no advice for a stable market. This quality criterion should be compromised by end user developed systems.

The **second** criterion was *decomposition* which is widely accepted as a design principle in software development, database design, and in knowledge-based systems development. Design rules, such as modular design, use of Object-Oriented programming concepts, or database normalization are examples of design concern.

Fundamentally, designers are advised to design systems by grouping associated components together (cohesion), and to separate those that do not belong together (decoupling). To measure decomposition, three metrics were used:

1. largest number of premises in a rule,
2. levels of reasoning within the knowledge base, and
3. The use of OR connectors.

The OR criterion might be perceived as less important for user developed applications, violations of rules of decomposition will limit growth and hamper maintenance, and therefore limit the use of the developed systems as an organizational asset.

The **third** criterion is the use of *probabilistic reasoning* within the knowledge base reflecting the capabilities of approximate reasoning, use of heuristics, and pursuit of multiple paths. As professionals in their field of expertise, the individuals participating in this study were aware of non-deterministic concepts. They had received formal instruction about heuristic reasoning and the mechanisms of probabilistic reasoning within knowledge-based systems. The above measures are summarized in figure (6).

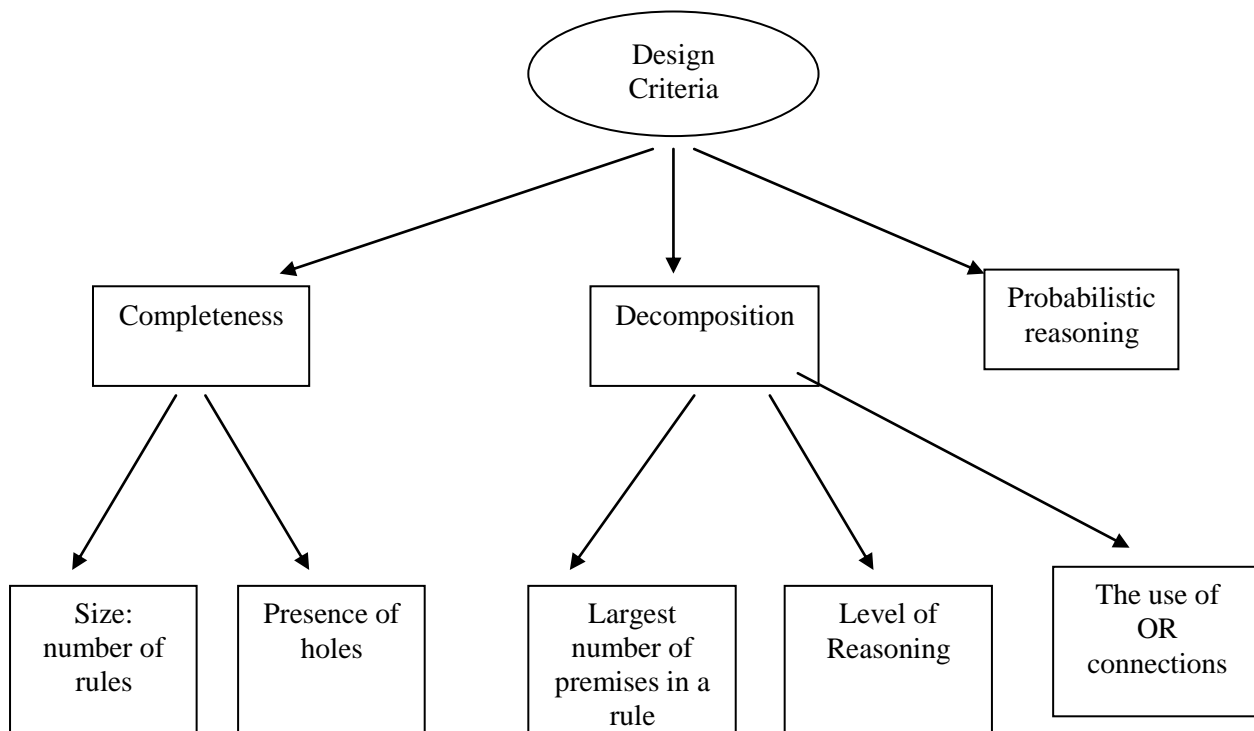


Figure 6: Design criteria measures

The study did not measure the content quality of each knowledge base. Thus, a knowledge base could “score highly” but at the same time be simple in its advice capability, if the developer chose such a direction. Knowledge bases were not allowed to be trivial, but there is clearly a range of content quality. It is impossible to prescribe the area study on participants because the purpose of the exercise is to let them build a knowledge base for the area they are highly interested in. Hence there is no uniformity [51] in the knowledge base content and no well justifiable way to assess content quality *per se*.

The above measures are not exhaustive ones. There are others not considered such as bugs that would not allow a knowledge base to execute. A minimum qualification for all systems is that they were executable. Overall then, the performance measures used in this study covered partially the two defect areas associated with the knowledge base view of software maintenance, *content* and *structure*. When the probabilistic reasoning criterion coupled with previous measures one would expect the study to provide a multifaceted insight into the quality and performance of user developed systems.

All 30 projects were evaluated according to the six measurement criteria. The results are listed in Table I, which shows an overall wide range of values for the six metrics (Adapted from Wagner [51]).

Table I: Summary of performance measures

Project No.	Application	Max no of Premises	Holes	Rules	Levels	Use of OR	Prob. Reasoning
1	GIS Analyzer	5	Yes	47	2	No	No
2	Help desk	4	No	57	3	No	No
3	Help desk	4	Yes	67	3	No	Yes
4	Help desk	3	No	39	3	Yes	No
5	PC maintenance	7	No	26	2	No	No

	advisor						
6	Oil exploration advisor	6	Yes	120	4	No	No
7	Equipment selection	5	Yes	44	3	Yes	Yes
8	Dining out selection guide	4	No	75	3	No	No
9	Employee stress advisor	4	Yes	86	4	No	No
10	Blackjack advisor	3	No	128	4	Yes	No
11	Travel advisor	4	No	83	3	Yes	Yes
12	Financial aid advisor	5	No	87	4	Yes	Yes
13	Software explanation	3	Yes	90	4	No	Yes
14	Information auditing	4	No	35	3	Yes	Yes
15	Insurance profiling	3	No	95	3	No	No
16	Intelligent tutors	8	Yes	37	2	Yes	Yes
17	Online teaching	3	No	175	5	No	Yes
18	Risk assessment	3	Yes	32	4	No	No
19	Computer configuration	6	No	240	5	No	Yes
20	Librarian System	6	No	109	3	Yes	Yes
21	Auction facilitator	5	Yes	155	2	No	No
22	Patient diagnosis System	7	No	30	2	No	Yes
23	Medical advisor	5	No	51	4	No	No
24	Help desk	4	No	33	3	No	Yes
25	Human Resource Manager	13	No	91	3	Yes	No
26	Risk management	3	No	175	5	No	Yes
27	Risk assessment	3	No	32	4	No	No
28	Help desk	5	Yes	154	4	No	No
29	Help desk	6	No	109	3	Yes	Yes
30	Help desk	5	Yes	167	5	Yes	No
Average		4.9	12	92.3	3.4	11	14
St. dev.		2		53.6	0.9		

Table I represents different level of sophistication and development effort. In total, performance concerns were detected in 80% of the projects. Twelve projects out of the 30 have missing rules, five provide strong indications of poor decomposition, and 14 lacked probabilistic reasoning. At the same time, knowledge bases were not insignificant in size, with an average of 92 rules each. Looking at table I, one might issue a question regarding the acceptable levels of performance. For some criteria, the answers are simple. Existence of holes, for instance, is not acceptable at all.

Although the systems developed in the exercise is of considerable size (average 92 rules), almost 40% had missing rules. This occurs, because the system does not contain a rule that governs the combination of conditions specified by the user. It is likely to expect it from an unfinished prototype. Holes are frequently due to an oversight, where part of a range of values is not covered. This is frequent mistake for end users who frequently do not use knowledge formalization techniques, such as representation in decision tables. Holes become even more likely and more difficult to detect when the system is not well decomposed and therefore every rule contains many premises. Other holes exist because the novice developer assumes that specific conditions or combinations of conditions just cannot occur and therefore can be ignored.

4.2 Flat Knowledge Base

A “flat” knowledge base, like a flat organization, has few layers. The number of layers or levels in a knowledge base can be determined as follows [54]. The top-level goal is at Level 0. Rules that will satisfy the goal are Level 1, rules that are called by Level 1 rules become Level 2 rules, and so on. At one point, there are no further calls to rules (the knowledge-based system then asks the user or consults a database).

The highest level of rules then describes the depth of the knowledge base. The minimum level of depth for a knowledge base is 1, it can be 0, if the knowledge base contains no rules at all. While flatness is desirable in modern organizations, it is not necessarily desirable in knowledge bases. Especially when the knowledge base gets larger, we should expect more than one or two levels of reasoning. In fact, none of the projects listed in Table 1 contains one level only. An obvious example of a flat knowledge base is for instance Project No. 24 with 133 rules and only 3 reasoning levels. A flat knowledge base is the result of poor decomposition of the application problem.

Using more rules typically means more development, maintenance, and more verification effort. This is where holes easily appear in the knowledge base. When the number of rules becomes large one would expect more slips. In the deep design, fewer rules have to be written. Writing fewer rules in each level will simplify verification process [35].

A third disadvantage is the lack of intermediate results. This disadvantage is based on a feature of knowledge-based systems, which usually remember intermediate reasoning results of an inference, regardless of the final outcome of the inference. The intermediate results can then be used for other inferences during the same session.

Given the disadvantages associated with flat knowledge bases, why would designers actually create them? Novice developers report that flat knowledge bases are easy to design. The developer sets up a “rule template” and then produces variants, often simply through “cut-and-paste” as the case with the project outlined in Table 1. This procedure can initially quickly produce a small knowledge base that deals with some of the relevant case scenarios. It is only later that the developer realizes how difficult it is to complete the knowledge base and to detect any existing errors in it. On the other hand, creating a deep knowledge base requires more initial insight and planning. The developer has to understand how to decompose the knowledge of the domain and then implement the knowledge in decomposed form. This of course requires more understanding of the design principles and highly developed skill [36].

4.3 Rule Coupling through OR

Some designers may consider the use of OR within the knowledge base rules an efficient mechanism to reduce the overall number of rules. The use of OR allows the combination of two sets or more premises that result in the same conclusions. In our set, eleven out of 30 projects used OR statements in rules. Premises are considered related [51, 53], if they refer to the same condition but differ in the value of the decision. Consider for example the rule: IF Economy = Highly Stable OR Economy = Stable THEN ...). It differs only by the value. The guidelines of programming recommend not to combining unrelated elements. In our study, this is considered coupling and its value was 11 as in column “Use of OR” in Table 1.

A representation without OR has further advantages if combined with probabilistic reasoning. When premises are separated, one can attach a certainty factor to the conclusion of each separated rule and increase the cumulative confidence factor for a conclusion as more and more evidence is gathered. When developers use OR, it is often not to shorten the knowledge base, but because inexperienced developers draw analogies to traditional programming where both AND and OR are common ones. If AND is used in knowledge-based systems, then why not also OR? It requires some experience for novice developers to realize that writing a new rule (with the same conclusion) is identical to attaching a premise with an OR, so that the use of OR is in fact unnecessary [30, 37].

4.4 Lack of Probabilistic Reasoning

End user developers often do not seem to realize that the knowledge they represent is probabilistic instead of “hard and fast”. As a result, rules that should be described using confidence factors are stated as certain. For example, the majority of projects (16 out of 30) listed in Table 1 was programmed in deterministic form despite the constant advice giving to considering probabilistic reasoning. Two reasons were given for this type of representation. First, based on their verbal comments, developers apparently truly believed their tasks were non-probabilistic [51, 52, 54], even when they were obviously not. Knowledge-based developers may recognize that the rules they know do not work all the time, and are comfortable in applying “back-up” rules.

At the same time, they seem to feel uncomfortable in operationalizing this method of reasoning in the form of most-likely rules with high probability values and less-likely rules with low probabilities. Second, being novices, the developers considered the introduction of meaningful confidence factors too difficult [46]. They simply found it difficult to attach a specific probability value to a rule. It should be noted that even experienced developers may find the assignment of factors representing the probabilistic nature of the knowledge-bases’ knowledge difficult. They typically see the problem as one of determining the most appropriate probability values (or certainty factors) so that final conclusions are presented with meaningful probability numbers [16].

5. Impact of Knowledge Base Development on Knowledge Management

From the analysis of our study one could conclude a number of shortcomings in the end users’ ability to construct quality knowledge. It has suggested a further inability to express key problem solving knowledge as well. The implications of both concerns are discussed below.

The projects analyzed were of considerable size and at the same time they exhibited a range of problem symptoms. Large size project is seemingly easier to generate than good design. Lack of probabilistic reasoning indicates that developers do not fully capture the essence of knowledge-based reasoning. Missing rules point to end user difficulties in the systematic validation of knowledge bases. Poor decomposition is evidence that developers ignored, willingly or unwillingly, knowledge base decomposition principles (thus complicating future maintenance and increasing the size of enhancement problems). End user knowledge-based systems might work at the present, but lack the features that allow them to easily grow and be maintainable in the future. These quality concerns were encountered even though all individuals had been previously given a awareness of the important of knowledge base decomposition, probabilistic reasoning, use of OR, and knowledge base verification. In addition, all projects had gone through revision iterations

With knowledge base design and maintenance being time-consuming tasks, there are obvious limits to the maximum size knowledge base a developer can create. The participants have spent in this exercise about 50 hours on average resulted in 92 rules on average. The size is in the range of systems developed at DuPont or Eastman Kodak [38]. The projects are considered modest ones compared with American Express’ Authorizer’s Assistant which contains about 850 rules, Coopers & Lybrand’s Knowledge-based contains about 3000 rules, DEC’s XCON system about 10,000 rules [39]. Also, the above professional systems contain well-written rules where one rule may represent knowledge captured by several poorly written ones.

The limits encountered in end user knowledge-based system development create a dilemma. On one hand, companies do not have the resources to use professional developers to codify significant amounts of end user knowledge. On the other hand, end users, while possessing the knowledge are unable to codify it themselves in a meaningful manner, using the available tools [36]. *Groupware* products, such as Lotus Notes are partly suitable to overcome the dilemma and enable end users recording knowledge in plain text. Knowledge in plain text is usually ambiguous, incomplete, and cannot be easily processed.

Despite the drawbacks outlined above, knowledge-based systems when interfaced to management systems assist in and enhance searching knowledge, help establish knowledge profiles of individual or groups, determine the relative importance of the contributed or the accessed knowledge, identify pattern in data, determine meaningful relationship, browsing, help in scanning various types of documents, help forecast future results from existing knowledge, and provide a natural language or voice command-driven user interface for the knowledge management systems[38].

This research investigated the impact of knowledge management on strategic orientation and organizational performance by using structural questioners. A number of managers and employees of certain production companies were received questioners regarding the impact of knowledge management on strategic orientation and firm performance. The results of the correlation coefficient showed that there is a significant positive correlation between knowledge management, strategic orientation and firm performance. The analysis also confirmed that the impact of knowledge management on strategic orientation and organizational performance is positive and significant. In addition, knowledge management has a significantly positive effect on firms' performance indirectly through strategic orientation. In general, the results of this investigation confirm the positive effect of knowledge management on strategic orientation and firms' performance [40].

Firms need to be aware of related knowledge issues because knowledge management as a business strategy, simultaneously, operates all over the firm, and it is considered as the tool advancement of a firm in overall program .In addition, the strategic direction of the firm in relation to customers, competitors and technology companies provides a basis to improve performance because it can identify customer needs and improve their satisfaction and cause using new technologies in production and provide the ability to rapidly respond to threats that endanger their companies [38]. There are a number of confirmations to the motivation leading firms to undertake a knowledge management effort. Typical considerations driving a knowledge management effort include [40, 41]:

- Making available increased knowledge content in the development and provision of products and services,
- Achieving shorter new product development cycles,
- Facilitating and managing innovation and organizational learning,
- Leveraging the expertise of people across the organization,
- Increasing network connectivity between internal and external individuals,
- Managing environments by enabling employees to obtain relevant insights and ideas appropriate to work,
- Solving intractable or wicked problems, and
- Managing intellectual capital and intellectual assets in the workforce, such as expertise and know-how possessed by key individuals, or stored in repositories.

The most recent advancement of knowledge-based systems and knowledge management has been to adopt the technologies for the development of systems that use the internet. The internet often has to deal with complex, unstructured data that can't be relied on to fit a specific data model. The technology of knowledge-based systems and especially the ability to classify objects on demand is ideal for such systems. The model for these kinds of knowledge-based Internet systems is known as the Semantic Web [42]. The term is broad, and is used to refer to many kinds of systems; examples include IBM's Watson[43] and the Wolfram Language [44].

Although knowledge-based systems (KBSs) have been proposed to support product development activities and new knowledge modeling methodologies have been developed, they are still far from complete. This area has become attractive to many researchers and as a result, many new knowledge-based systems, methods and tools have been developed. However, to the best of our knowledge, knowledge-based systems for product development have not been systematically reviewed, compared and summarized [45].

6 Conclusions

Knowledge requires the storage and processing of associations through which meaning can be derived from the information. Association can be represented in explicit and observable forms in a knowledge base. The paper discusses issues relating to design a better quality interactive interface system for human to have a dialog with the knowledge management systems. At the same time, the paper investigates the ease of evaluation and implementation of a knowledge management system. It performs a major role in providing users with capabilities of dealing with underlying systems. Designing a good interface style using knowledge bases can have a profound effect on the nature of the dialog.

Many organizations have recognized the implication of intelligent information technologies to support performing their daily activities and more effective ways. Creating intelligent systems on a large scale remain a difficult Endeavour and all famous intelligent information systems have been the product of costly professional development efforts. This study indicated that there was little chance that similar successes can be achieved through end user developed systems. End user development will be limited in content, quality, and size and will not scale up. Companies that look for new ways to create and manage knowledge will have to search for alternative means to achieve this goal. Intelligent interface to knowledge-based systems is one of the promising technologies to enhance performance.

Interfacing a knowledge base oriented system can be seen as a dialog between the knowledge base and the user. It plays a major role in providing users with capabilities dealing with underlying systems. Designing a good interface style can have a profound effect on the nature of the dialog. Design of a user interface involves determining approaches in which users interact with the knowledge-based system. This study investigates the improvement of performance by performing an actual study of thirty experimental systems developed specially for the sake of the study. Results confirm that a good interface has a great impact on the performance of knowledge management systems.

The study confirms that a good intelligent interface will enhance the understandability of users and provides them with a knowledgeable background. Users who learn to describe their knowledge in a format similar to rules (instead of cases) and who learn to decompose the knowledge, into different concepts and different levels of reasoning, might provide a much better knowledge base to other organization members.

References

1. Frederick, H-R; Waterman D. and Lenat D., (1983). Building Expert Systems. Addison-Wesley.
2. Davenport, T. H., (1994). "Saving IT's Soul: Human Centered Information Management". Harvard Business Review 72 (2), pp. 119–131.
3. www.unc.edu, (2016). "Introduction to Knowledge Management". University of North Carolina at Chapel Hill. Archived from the original on March 19, 2007. Retrieved 1 June 2016.
4. Rostami, N. A. (2014). Integration of Business Intelligence and Knowledge Management: A literature review, Journal of Intelligence Studies in Business Vol. 4, No. 2, PP. 30-40.
5. Karna N., Supriana I. and Maulidevi U., (2014). Intelligent Interface for a Knowledge-based System, TELKOMNIKA Vol. 12, No. 4, December 2014, pp. 1096 – 1104.
6. Dix A., Finlay J., Abowd G. D. and Beale R., (2004). Human Computer Interaction, Third Edition, Prentice-hall.
7. Karna N., Supriana I. and Maulidevi U., (2014). Intelligent Interface for a Knowledge-based System, TELKOMNIKA, Vol.12, No.4, December 2014, pp. 1096-1104.

8. Russel S. and E. Wefald (1991). *Do the Right Thing: Studies in Limited Rationality*. MIT Press, Cambridge, Mass, 1991.
9. Bennet A., and A. Bennet, (2003). "The Partnership Between Organizational Learning and Knowledge Management", Chapter 23 in C.W. Hollsapple (ed.) *Handbook of Knowledge Management: Knowledge Matters*, Vol. 1, Heidelberg: Springer-Verlag, 2003.
10. Siemens AG (2016). Retrieved from https://www.siemens.com/annual/14/en/download/pdf/Siemens_AR2014_SupervisoryBoard-ManagingBoard.pdf, July 2016.
11. Vasilash G.S., "447,000 Heads are better than one", *Automotive Design and production*, June 2002.
12. Turban E., J. E. Aronson, and T. Liang, "Decision Support Systems and Intelligent Systems, Prentice-Hall, 7th Edition, 2005.
13. Smith H., and J. Mckeen, "Creating and facilitating communities of practice", Chapter 20 in C.W. Hollsapple (ed.) *Handbook of Knowledge Management: Knowledge Matters*, Vol. 1, Heidelberg: Springer-Verlag, 2003.
14. Wang, S. and Noe, R. A. (2010). Knowledge sharing: A review and directions for future research", *Human Resource Management Review*, Vol. 20, No. 2, pp 115-131.
15. Introduction to Knowledge Management". www.unc.edu. University of North Carolina at Chapel Hill. Archived from the original on March 19, 2007. Retrieved 11 September 2014.
16. Alavi M., and D. Leidner, (2001). "Knowledge Management and Knowledge Management Systems: Conceptual foundation and an agenda for Research", *MIS Quarterly*, 2001, pp 107-136
17. Cingular Wireless, (2016). www.cingular.com, accessed May 2016.
18. O' Herron J., (2003). "Building the Bases of knowledge", *Call Center Magazine*, January 2003.
19. Hinds R. S., and J. E. Aronson, (2002). "Developing the Requisite organizational, Attitudinal, and Behavioral Conditions for Effective Knowledge Management", proceeding of the American Conference of Information Systems, U.S.A, 2002
20. Vaas L. "Brainstorming, (1999). *PCWeek*, vol.16, No.22, May 1999.
21. Davenport, D., and M. Sena, (2003). "Technologies for knowledge Derivation", Chapter 40 in C.W. Hollsapple (ed.) *Handbook of Knowledge Management: Knowledge Matters*, Vol. 1, Heidelberg: Springer-Verlag, 2003.
22. Surbakti, H., (2015). Integrating Knowledge Management and Business Intelligence Processes for Empowering Government Business Organizations, *International Journal of Computer Applications* 114(5):March 2015,
23. Luo X., Zhang J.,Ye F., Wang P. and Cai C.,(2014). Power Series Representation Model of Text Knowledge Based on Human Concept Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 44, No. 1, pp. 86-102.
24. Shuchao Wan, Jun Wei, Jingyu Song, Heqing Guan. Developing a Selection Model for Interactive Web Services. *IEEE International Conference on Web Services*. Chicago. 2006: 231-238.
25. Jun Wang, Li Zou, Hong Peng, Gexiang Zhang. An Extended Spiking Neural P System for Fuzzy Knowledge Representation. *International Journal of Innovative Computing, Information and Control, ICIC International*. 2011; 7(7(A)): 3709–3724.
26. Craig Schlenoff, Sebti Foufou, Stephen Balakirsky. Performance Evaluation of Robotic Knowledge Representation (PERK). *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. Maryland. 2012: 1-8.
27. Zeti Darleena Eri, Rusli Abdullah, Marzanah A. Jabar, Masrah Azrifah, Azmi Murad. Ontology-based Knowledge Model for Virtual Communities Profile. *Malaysian Conference in Software Engineering (MySEC)*. Johor Baru. 2011: 508-511.
28. Chung-Yi Huang and Rung-Ching Chen. Theses Cluster Based on Bilingual and Synonymous Keyword sets Using Mutual Information. *Proceedings of the Eight International Conference on Machine Learning and Cybernetics*. Baoding. 2009: 2999-3004.

29. Maes, P., (1994). "Agents that reduce work and information overload", Communications of the ACM 37(7), 1994.
30. Panko, R. (1998). What We Know About Traditional end user application Errors. *Journal of End User Computing*, 10, 1998. pp. 15-21.
31. Methlie L.B. (1993). "Performance Modeling: A Cognitive Approach to Building Knowledge-based Systems, in P.R. WATKINS and Eliot, L.B. (ed.) chapter 7 Expert Systems in Business and Finance: Issues and applications, John Wiley, 1993.
32. D. J. Power, R. Sharda, and F. Burstein (2015), Decision Support Systems, Vol. 7., Published online.
33. McCall J.A., Richards, P.K., & Walters. G.F. (1997). Factors in Software Quality Assurance. *RADC-TR-77-369*, Rome Air Development Center, Rome, 1977.
34. M. Suwa, A. C. Scott, E. H. Shortliffec, (1982). An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System, *AI Magazine* Volume 3 Number 4.
35. Delgrande J., Dubois D., and Lang J., (2006). Iterated revision as prioritized merging. *Proc. of KR06*:201-220.
36. Benferhat, Dubois D. and Prade H., (1993). Argumentative Inference in Uncertain and inconsistent Knowledge Bases, pp. 411-419, in *Proceedings of the Ninth Conference of Uncertainty in Artificial Intelligence*, Edited by Heckerman D. and Mamdani A., Morgan Kaufmann Publishers, Inc., U.S. A.
37. Blili S., Raymond L. & Rivard S., (1996). Definition of End User Computing Sophistication, *Journal of End User Computing*, Vol. 8, No. 2, pp. 3-10.
38. King, W.R. 2008. An integrated architecture for the effective knowledge organization. *Journal of Knowledge Management* 12(2): 1367–1380.
39. Cooper M. A. and Leidner, D. E. (2001). "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues". *MIS Quarterly* 25 (1): 107–136.
40. Gupta B., L. Iyer, and J.E. Aronson, (2000). "knowledge management: A Taxonomy, practices and challenges", *Industrial Management and Data systems*, vol. 100, No 1 and 2, 2000.
41. Berners-Lee, T.; J. Hendler; O. Lassila, (2001). "The Semantic Web A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities". *Scientific American* **284**, pp. 34–43.
42. "What is IBM Watson?", (2016). Retrieved from www.ibm.com, July 2016.
43. "Wolfram Language for Knowledge-Based Programming", (2016). Retrieved from www.wolfram.com, July 2016.
44. Li B. M., Xie S. Q. and Xu X., (2011). Recent development of knowledge-based systems, methods and tools for One-of-a-Kind Production, *Knowledge-Based Systems*, Volume 24, Issue 7, October 2011, pp. 1108–1119.
45. L. Nedović and V. Devedžić, (2002). Expert Systems in Finance – A Cross-Section of the Field, Retrieved from <http://devedzic.fon.rs/publications/ESwA2002-1.pdf>, July 2016.
46. Seraji R. , Shoar S. M. A. , Abasi A. and Norouzi D., (2013). Study of the relationship between knowledge management and organizational culture Case study: agriculture organization of Qom, *International Research Journal of Applied and Basic Sciences*, Vol. 4, No. 6, pp. 1535-1542.
47. Botha, A., Kourie, D., & Snyman, R. (2008). *Coping with Continuous Change in the Business Environment*, Knowledge Management and Knowledge Management Technology, Chandice Publishing Ltd.
48. Holsappl C. W. and Whinston A. B., (1996). *Decision Support Systems: A Knowledge Based Approach*, St. Paul: West.

49. Edwards J. S. (Editor), (2015). The Essential of Knowledge Management, Retrieved from <https://books.google.iq/>, July 2016.
50. Adelman L. and Riedel S. L. (1997). Handbook for Evaluating Knowledge-Based Systems: Conceptual Framework and Compendium of Methods, Published by Springer Science + Business Media, LLC.
51. Wagner C., (2002). End Users as Expert System Developers? - Chapter 3 in: Barrier T., (2002). Human Computer Interaction Development & Management, Published by Idea Group Inc.
52. Date C. J., (1990). An Introduction to Database Systems (5th edition), Reading: Addison-Wesley.
53. Mallach E. G., (1994). Understanding Decision Support Systems and Expert Systems, Boston: McGraw-Hill.
54. Olson D. L., Schellenberger R. E. & Mechitov A. I. (1995). Teaching Knowledge Base Consistency and Completeness, Journal of Computer Information Systems, pp. 7-12.