# EXPERT SYSTEM TECHNOLOGY pdf

## 1: Expert System: Artificial Intelligence: Cognitive Computing Company

*Effective data and information management is the foundation for competitive advantage for enterprises in any industry. Today's information-infused world has added another essential component for differentiation: cognitive technology.*

One of the existing reasons for this disparity is the lack of motivation of learning systems which are not significantly oriented to expertism in a given field. The momentum of the progress in the technologically advanced countries is such that the technological gap between the developed and developing countries is widening overall. Developing societies are being left behind because of their slow rate of absorption of technology. Before they can properly digest the inventions of the day, many more new inventions have been added. This is a step forward towards disintegrating the existing expertise and experts. Experts and expertise are identified differently. Presently the technology gives users the choice of decision according to the requirement as to whether he wishes to hire expertise or experts. Nations competing in technology may look forward to the expertise in a suitable and acceptable form. It is proposed to review how this higher technology area can help to overcome the dearth of experts by acquiring the relevant expertise for complicated problem handling in the cross section of our society. Through experimental successes attempting efforts geared to the development of Artificial Intelligence AI has successfully emerged2 the emulated production of human-like expertise which may be called pseudo-expertise. More or less in the last three decades, laboratory curiosities of applied artificial intelligence have evolved into technical and commercial efforts. This system employs computers in ways that differ markedly from the conventional data-processing applications, which are algorithm-oriented, and instead they can: Unlike the conventional systems, which deal with rigidly structured procedures, expert systems can weigh evidence, cope with judgement and uncertainty and can explain their recommendations. They can also be improved, as knowledge is refined. All this is because the computer can embody data, knowledge and inference mechanism which a single person, howsoever capable, just cannot possess. Human expertise is essential to any organizational functioning. Humans even experts can, on occasion, forget details which may be vital. Expert systems do not fail to ask relevant questions or absent-mindedly ignore important factors which may be up for discussion. Expert systems, moreover are consistent. Given the same input, the Expert system consistently reaches the intended conclusions. Expert systems are always available and can be interrogated by many users at once. Human experts often have to travel to deploy their knowledge, and are restricted by time and other business pressures. Expert systems can be copied and distributed whenever there is a suitable computer. The earlier phase of information technology faced considerable hesitation due to the fear of being job killers. The add-on features of this technology, instead of acquiring the automation of clerical activity, is now targeting at the automation of the intellectual components of task performance. By successively acquiring this technology, the developing countries should be ready to make available the expert consultancies services at large. In order to avoid waste in money and efforts, the new technology, should be acquired along well-defined and prescribed lines, and should be dedicated to the furtherance of national interests, As the effects of the new technology acquiring process would be felt along a wide spectrum of the society as a whole, the objectives must be set and given sets of priorities, A mechanism for monitoring the tracks of acquisition, prescribing technology standards, the suitability of products developed for marketing and sales, and checking the flooding of the market with sub-standard or unmarketable, goods, has to be evolved. This approach embodies a check similar to the one, which is applied in the printing industry. As the impact of the new technology is bound to be tremendous, it needs checks and balances in a technology based society. With developments in information technology, new possibilities are emerging and new threats are being perceived. Some of the other implicit attempts of the Expert Systems are: Is an attempt undesirable? This would go to affect the services and the type of services the society would be getting from the human experts. Some of the vital service areas would be in the field of medical diagnoses, and follow-up treatments. In such situations there could be manifold impacts. Sociologists are rather wary about the way in which the new technology will make its impact felt upon society. This, in itself, might lead to increased or decreased creativity, depending upon the society as a whole or the individual

concerned. Some of these are: Medical information logic, to diagnose blood and meningitis infection and advise physicians about antibiotic therapy. Consultant system for mineral exploration. It can resolve ambiguity by assigning preference rating to different paraphrases. A German language interface, specially tailored to database access in a number of areas. Also an interactive translation system, finding unknown word or words with ambiguous meaning. It converts, alphanumeric text into human quality speech. To advice engineers how to use a complex structural analysis programme. A variety of applications falls under the scope of applied AI and expert systems. The guidelines which can help in selecting a problem for knowledge-engineering exposure and success of application are: Two of the basic aims are: They must support the communication requirements and communication behaviour of a society. As the main carrier of human history, culture, and civilization, the language must be able to cover all intellectual and social spheres of the society. Languages which cannot fulfill these two requirements will sooner pr later fade away or will be partly replaced by other living languages. The written form of the Japanese language is not amenable to programming. Programmers would be able to work much more efficiently, if they could, in their own language. This argument also applies to Urdu. Urdu script has its own characteristics. Some are quite different from English sinistral type of language writing. Some of the bottle-necks which need re-solution are:

## 2: Cognitive Technology Software: Cogito

*Expert system, a computer program that uses artificial-intelligence methods to solve problems within a specialized domain that ordinarily requires human expertise. The first expert system was developed in by Edward Feigenbaum and Joshua Lederberg of Stanford University in California, U.S. Dendral, as their expert system was later known, was designed to analyze chemical compounds.*

For such AI systems every effort is made to incorporate all the information about some narrow field that an expert or group of experts would know, so that a good expert system can often outperform any single human expert. There are many commercial expert systems, including programs for medical diagnosis , chemical analysis , credit authorization, financial management, corporate planning, financial document routing, oil and mineral prospecting, genetic engineering , automobile design and manufacture, camera lens design, computer installation design, airline scheduling, cargo placement, and automatic help services for home computer owners. Knowledge and inference The basic components of an expert system are a knowledge base , or KB, and an inference engine. The information to be stored in the KB is obtained by interviewing people who are expert in the area in question. Rules of this type are called production rules. The inference engine enables the expert system to draw deductions from the rules in the KB. Some expert systems use fuzzy logic. In standard logic there are only two truth values, true and false. This absolute precision makes vague attributes or situations difficult to characterize. When, precisely, does a thinning head of hair become a bald head? The substance to be analyzed might, for example, be a complicated compound of carbon , hydrogen , and nitrogen. MYCIN would attempt to diagnose patients based on reported symptoms and medical test results. The program could request further information concerning the patient, as well as suggest additional laboratory tests, to arrive at a probable diagnosis, after which it would recommend a course of treatment. Using about production rules, MYCIN operated at roughly the same level of competence as human specialists in blood infections and rather better than general practitioners. Nevertheless, expert systems have no common sense or understanding of the limits of their expertise. Expert systems can also act on absurd clerical errors, such as prescribing an obviously incorrect dosage of a drug for a patient whose weight and age data were accidentally transposed. The project began in under the auspices of the Microelectronics and Computer Technology Corporation, a consortium of computer, semiconductor , and electronics manufacturers. The most ambitious goal of Cycorp was to build a KB containing a significant percentage of the commonsense knowledge of a human being. Millions of commonsense assertions, or rules, were coded into CYC. With only a fraction of its commonsense KB compiled, CYC could draw inferences that would defeat simpler systems. Among the outstanding remaining problems are issues in searching and problem solvingâ€”for example, how to search the KB automatically for information that is relevant to a given problem. AI researchers call the problem of updating, searching, and otherwise manipulating a large structure of symbols in realistic amounts of time the frame problem. Some critics of symbolic AI believe that the frame problem is largely unsolvable and so maintain that the symbolic approach will never yield genuinely intelligent systems. It is possible that CYC, for example, will succumb to the frame problem long before the system achieves human levels of knowledge. Connectionism Connectionism, or neuronlike computing, developed out of attempts to understand how the human brain works at the neural level and, in particular, how people learn and remember. In the neurophysiologist Warren McCulloch of the University of Illinois and the mathematician Walter Pitts of the University of Chicago published an influential treatise on neural nets and automatons , according to which each neuron in the brain is a simple digital processor and the brain as a whole is a form of computing machine. They were able to train their networks to recognize simple patterns. The simple neural network depicted in the figure illustrates the central ideas of connectionism. Each of the neurons is either firing 1 or not firing 0. For example, suppose that only two of the input neurons, X and Y, are firing. Since the weight of the connection from X to N is 1. As shown in the figure, N has a firing threshold of 4. A section of an artificial neural networkIn the figure the weight, or strength, of each input is indicated by the relative size of its connection. The firing threshold for the output neuron, N, is 4 in this example. Hence, N is quiescent unless a combination

of input signals is received from W, X, Y, and Z that exceeds a weight of 4. Training the network involves two steps. First, the external agent inputs a pattern and observes the behaviour of N. Second, the agent adjusts the connection weights in accordance with the rules: If the actual output is 0 and the desired output is 1, increase by a small fixed amount the weight of each connection leading to N from neurons that are firing thus making it more likely that N will fire the next time the network is given the same pattern ; If the actual output is 1 and the desired output is 0, decrease by that same small amount the weight of each connection leading to the output neuron from neurons that are firing thus making it less likely that the output neuron will fire the next time the network is given that pattern as input. The external agentâ€"actually a computer programâ€"goes through this two-step procedure with each pattern in a training sample, which is then repeated a number of times. During these many repetitions, a pattern of connection weights is forged that enables the network to respond correctly to each pattern. The striking thing is that the learning process is entirely mechanical and requires no human intervention or adjustment. The connection weights are increased or decreased automatically by a constant amount, and exactly the same learning procedure applies to different tasks. He made major contributions to the field of AI, both through experimental investigations of the properties of neural networks using computer simulations and through detailed mathematical analysis. Rosenblatt was a charismatic communicator, and there were soon many research groups in the United States studying perceptrons. Rosenblatt and his followers called their approach connectionist to emphasize the importance in learning of the creation and modification of connections between neurons. Modern researchers have adopted this term. The method, with substantial improvements and extensions by numerous scientists, and the term back-propagation are now in everyday use in connectionism. Conjugating verbs In one famous connectionist experiment conducted at the University of California at San Diego published in , David Rumelhart and James McClelland trained a network of artificial neurons, arranged in two layers of neurons, to form the past tenses of English verbs. Root forms of verbsâ€"such as come, look, and sleepâ€"were presented to one layer of neurons, the input layer. A supervisory computer program observed the difference between the actual response at the layer of output neurons and the desired responseâ€"came, sayâ€"and then mechanically adjusted the connections throughout the network in accordance with the procedure described above to give the network a slight push in the direction of the correct response. About different verbs were presented one by one to the network, and the connections were adjusted after each presentation. This whole procedure was repeated about times using the same verbs, after which the network could correctly form the past tense of many unfamiliar verbs as well as of the original verbs. For example, when presented for the first time with guard, the network responded guarded; with weep, wept; with cling, clung; and with drip, dripped complete with double p. This is a striking example of learning involving generalization. Sometimes, though, the peculiarities of English were too much for the network, and it formed squawked from squat, shipped from shape, and membled from mail. Another name for connectionism is parallel distributed processing , which emphasizes two important features. First, a large number of relatively simple processorsâ€"the neuronsâ€"operate in parallel. Second, neural networks store information in a distributed fashion, with each individual connection participating in the storage of many different items of information. The know-how that enabled the past-tense network to form wept from weep, for example, was not stored in one specific location in the network but was spread throughout the entire pattern of connection weights that was forged during training. The human brain also appears to store information in a distributed fashion, and connectionist research is contributing to attempts to understand how it does so. Other neural networks Other work on neuronlike computing includes the following: Networks can recognize faces and other objects from visual data. A neural network designed by John Hummel and Irving Biederman at the University of Minnesota can identify about 10 objects from simple line drawings. The network is able to recognize the objectsâ€"which include a mug and a frying panâ€"even when they are drawn from different angles. Networks investigated by Tomaso Poggio of MIT are able to recognize bent-wire shapes drawn from different angles, faces photographed from different angles and showing different expressions, and objects from cartoon drawings with gray-scale shading indicating depth and orientation. Neural networks are able to convert handwritten and typewritten material to electronic text. Internal Revenue Service has commissioned a neuronlike system that will automatically read tax returns and

correspondence. Neural networks also convert speech to printed text and printed text to speech. Neural networks are being used increasingly for loan risk assessment , real estate valuation, bankruptcy prediction, share price prediction, and other business applications. Medical applications include detecting lung nodules and heart arrhythmias and predicting adverse drug reactions. Telecommunications applications of neural networks include control of telephone switching networks and echo cancellation in modems and on satellite links. Nouvelle AI distances itself from strong AI, with its emphasis on human-level performance, in favour of the relatively modest aim of insect -level performance. Practitioners of nouvelle AI assert that true intelligence involves the ability to function in a real-world environment. For example, a robot whose simple behaviours include collision avoidance and motion toward a moving object will appear to stalk the object, pausing whenever it gets too close. Herbert searches desks and tables for empty soda cans, which it picks up and carries away. More recently, Brooks has constructed prototypes of mobile robots for exploring the surface of Mars. See the photographs and an interview with Rodney Brooks. Herbert, the robot, c. After collecting an empty can with its robotic arm, Herbert would return it to a recycling bin. To see a larger image and obtain information on each robot, click on the individual photograph. Genghis weighs about 1 kilogram 2. Attila, like its predecessor Genghis, is a small, six-legged robot, but, whereas Genghis has no independent power source, Attila was equipped with solar cells to recharge its batteries. With its arm attached, Pebbles can collect samples or handle dangerous objects. Nouvelle systems do not contain a complicated symbolic model of their environment. A nouvelle system refers continuously to its sensors rather than to an internal model of the world: As Brooks insisted, the world is its own best modelâ€”always exactly up-to-date and complete in every detail. The situated approach Traditional AI has by and large attempted to build disembodied intelligences whose only interaction with the world has been indirect CYC, for example. Nouvelle AI, on the other hand, attempts to build embodied intelligences situated in the real worldâ€”a method that has come to be known as the situated approach. Brooks quoted approvingly from the brief sketches that Turing gave in and of the situated approach. He advocated that both approaches be pursued, but until recently little attention has been paid to the situated approach. The situated approach was also anticipated in the writings of the philosopher Bert Dreyfus of the University of California at Berkeley. Beginning in the early s, Dreyfus opposed the physical symbol system hypothesis , arguing that intelligent behaviour cannot be completely captured by symbolic descriptions. As an alternative , Dreyfus advocated a view of intelligence that stressed the need for a body that could move about, interacting directly with tangible physical objects. Once reviled by advocates of AI, Dreyfus is now regarded as a prophet of the situated approach. Critics of nouvelle AI point out the failure to produce a system exhibiting anything like the complexity of behaviour found in real insects. Suggestions by researchers that their nouvelle systems may soon be conscious and possess language seem entirely premature. Is strong AI possible? The ongoing success of applied AI and of cognitive simulation, as described in the preceding sections of this article, seems assured.

## 3: Expert system - The Full Wiki

*Expert System allows enterprises to stay competitive in a world that requires ever faster processing of increasingly diverse, high volume information. We do this through intelligent technology and applications that provide an accurate, automatic and immediate understanding of text.*

The methods and techniques used to build these programs are the outcome of efforts in a field of computer science known as Artificial Intelligence AI. Our purpose in writing this description of expert systems is to demystify these programs by revealing their surprisingly simple architecture, explaining a method by which they can be constructed inexpensively, and illustrating, using a simple example, the process by which they seem to reason. Distinguishing Features In conventional computer programs, problem-solving knowledge is encoded in program logic and program-resident data structures. Expert systems differ from conventional programs both in the way problem knowledge is stored and used. An expert system is a computer program, with a set of rules encapsulating knowledge about a particular problem domain i. These rules prescribe actions to take when certain conditions hold, and define the effect of the action on deductions or data. The expert system, seemingly, uses reasoning capabilities to reach conclusions or to perform analytical tasks. Expert systems that record the knowledge needed to solve a problem as a collection of rules stored in a knowledge-base are called rule-based systems. Utility of Expert Systems Expert systems are especially important to organizations that rely on people who possess specialized knowledge of some problem domain, especially if this knowledge and experience cannot be easily transferred. Artificial intelligence methods and techniques have been applied to a broad range of problems and disciplines, some of which are esoteric and others which are extremely practical. One of the early applications, MYCIN, was created to help physicians diagnose and treat bacterial infections. Expert systems have been used to analyze geophysical data in our search for petroleum and metal deposits e. They are used by the investments, banking, and telecommunications industries. They are essential in robotics, natural language processing, theorem proving, and the intelligent retrieval of information from databases. They are used in many other human endeavors which might be considered more practical. Rule-based systems have been used to monitor and control traffic, to aid in the development of flight systems, and by the federal government to prepare budgets. Advantages of Rule-Based Systems A rule-based, expert system maintains a separation between its Knowledge-base and that part of the system that executes rules, often referred to as the expert system shell. The system shell is indifferent to the rules it executes. This is an important distinction, because it means that the expert system shell can be applied to many different problem domains with little or no change. It also means that adding or modifying rules to an expert system can effect changes in program behavior without affecting the controlling component, the system shell. The language used to express a rule is closely related to the language subject matter experts use to describe problem solutions. When the subject matter expert composes a rule using this language, he is, at the same time, creating a written record of problem knowledge, which can then be shared with others. Thus, the creation of a rule kills two birds with one stone; the rule adds functionality or changes program behavior, and records essential information about the problem domain in a human-readable form. Knowledge captured and maintained by these systems ensures continuity of operations even as subject matter experts i. Furthermore, changes to the Knowledge-base can be made easily by subject matter experts without programmer intervention, thereby reducing the cost of software maintenance and helping to ensure that changes are made in the way they were intended. Rules are added to the knowledge-base by subject matter experts using text or graphical editors that are integral to the system shell. The simple process by which rules are added to the knowledge-base is depicted in Figure 1. Finally, the expert system never forgets, can store and retrieve more knowledge than any single human being can remember, and makes no errors, provided the rules created by the subject matter experts accurately model the problem at hand. Expert System Architecture An expert system is, typically, composed of two major components, the Knowledge-base and the Expert System Shell. The Knowledge-base is a collection of rules encoded as metadata in a file system, or more often in a relational database. The Expert System Shell is a problem-independent component housing facilities for

creating, editing, and executing rules. A software architecture for an expert system is illustrated in Figure 2. The shell portion includes software modules whose purpose it is to, Process requests for service from system users and application layer modules; Support the creation and modification of business rules by subject matter experts; Translate business rules, created by a subject matter experts, into machine-readable forms; Execute business rules; and Provide low-level support to expert system components e. Client Interface logic routes these requests to an appropriate shell program unit. For example, when a subject matter expert wishes to create or edit a rule, they use the Client Interface to dispatch the Knowledge-base Editor. Other service requests might schedule a rule, or a group of rules, for execution by the Rule Engine. It provides facilities that enable a subject matter expert to compose and add rules to the Knowledge-base. They must first be converted from their human-readable form into a form that can be interpreted by the Rule Engine. Converting rules from one form to another is a function performed by the Rule Translator. The translation of rules in their original form to a machine-readable form requires parsing the textual representation to obtain a data structure referred to as an Abstract Syntax Tree AST. The AST representation of rules is the memory-resident data structure that guides execution of the inference engine when it comes time to apply a rule. The AST is an abstract data type designed to make its interpretation by the Rule Engine simple and efficient. This abstract data type is very expressive and permits the construction of very complex and powerful rules. There is a third form in which rules may be expressed. A rule AST is converted into an equivalent form suitable for storage in the Knowledge-base. The way in which this information appears in the Knowledge-base depends on the storage technology. Relational databases provide an especially convenient and efficient means for storing metadata. The metadata corresponding to an AST in this case would reside in a collection of tables. The specific scheme chosen for saving the metadata must permit the ASTs to be rapidly reconstructed using database queries. The diagram in Figure 3 summarizes the several conversion operations the Rule Translator must perform as it adds rules to the Knowledge-base and as it retrieves rules from the Knowledge-base for execution. In the AST representation, this portion of the rule would be expressed as a binary tree, like the one in Figure 4, whose nodes are either arithmetic operators or operands. Once created, AST representations are converted into rule metadata and stored in the Knowledge-base. Rule metadata is simply a compact representation of ASTs. The role of the Rule Translator in the rule editing process is illustrated in the data flow diagram of Figure 5. In translating rules from one form to another, the structure of the original rule is never lost. It is always possible to recreate a human-readable rule, exactly, from its Knowledge-base representation or from its AST representation. It retrieves rules from the Knowledge-base, converts them to ASTs, and then provides them to its rule interpreter for execution. The Rule Engine interpreter traverses the AST, executing actions specified in the rule along the way. This process is depicted in Figure 6. Constructing an Expert System The construction of an expert system is less challenging than one might think, given the almost magical powers attributed to this class of programs. The task is made easier because, Large portions of the Rule Translator can be generated automatically using lexical analyzer and parser generators, and Text editors e. The design and the construction of the expert system involves the four major steps depicted in Figure 7. Design a Rule Language The surest way to make certain the expert system is well-received is to design a rule language that is easy for users to learn and easy to use. The language must serve the needs of the subject matter expert. Subject matter experts need a language that mimics the way they speak, think, and operate. Its lexical elements and syntax rules must make the resulting language appear as natural i. These goals must be balanced with the need to create a language that can be recognized by translators created with conventional translation writing tools. Rules must be expressed using a simple grammar, free of the idiosyncrasies of natural languages. Every sentence of the language used to express rules must have precise and unvarying meanings. A natural language with its inherent ambiguities is difficult to translate. This is because, in a natural language, words do not have a single meaning, nor are word meanings precise. In a natural language the meaning of words and phrases are colored by their context. Machine translation of natural languages is possible, but the program logic required is complex, expensive, and not wholly reliable. Machine translation of natural languages relies largely on heuristics that make translations imperfect. On the other hand, context-free languages can be made to appear natural-like. However, as any programmer knows, these languages are unforgiving of spelling and grammar

errors. Fortunately, methods for the translation of context-free languages are rooted in a mature and well-understood mathematics. Tools for building context-free languages translators make their construction straightforward and reliable. The tools used to generate translators for context-free languages require precise language definitions. These language definitions are called grammars. A sample grammar defining a hypothetical rule language appears in Figure 8. The grammar defines both the syntax and the semantics of the language. This production is read: A simple, but meaningless rule, conforming to our hypothetical grammar appears in Figure 9. Generate Translator from Rule Grammar The rules composed by subject matter experts must be translated and stored in the Knowledge-base before they can be executed. Rules are transformed into machine-readable forms by the Rule Translator. The Rule Translator is generated automatically from lexical specifications and from the rule grammar. Lexical specifications define the lexical units e. The rule grammar defines the syntax and semantics of the language. Figure 10 illustrates the method by which the Rule Translator is automatically generated. The lexical analyzer generator depicted in this diagram operates on a definition of the languages lexical units to produce a software component referred to as a lexical analyzer or scanner. The scanner reads and converts rules in text form into a sequence of lexical units called tokens. Each token is a sequence of logically related characters having special meaning. For example, the text, A stitch in time saves nine. The parser generator produces two software components, a Rule Parser and a collection of semantic action subprograms. The Rule Parser, receives token streams from the Scanner. It structures these tokens into meaningful units e. The parser invokes semantic actions i. Semantic actions are software units that are dispatched by the parser as it recognizes language constructs in the rule.

# EXPERT SYSTEM TECHNOLOGY pdf

## 4: Chp 1: The Applications Of Expert Systems

*Expert System Technology. There are several levels of ES technologies available. Expert systems technologies include âˆ’ Expert System Development Environment âˆ’ The ES development environment includes hardware and tools.*

The applications find their way into most areas of knowledge work. They are as varied as helping salespersons sell modular factory-built homes to helping NASA plan the maintenance of a space shuttle in preparation for its next flight. Applications tend to cluster into seven major classes. Diagnosis and Troubleshooting of Devices and Systems of All Kinds This class comprises systems that deduce faults and suggest corrective actions for a malfunctioning device or process. Medical diagnosis was one of the first knowledge areas to which ES technology was applied for example, see Shortliffe , but diagnosis of engineered systems quickly surpassed medical diagnosis. There are probably more diagnostic applications of ES than any other type. The diagnostic problem can be stated in the abstract as: This class has great commercial potential, which has been recognized. Examples involve airline scheduling of flights, personnel, and gates; manufacturing job-shop scheduling; and manufacturing process planning. Configuration of Manufactured Objects from Subassemblies Configuration, whereby a solution to a problem is synthesized from a given set of elements related by a set of constraints, is historically one of the most important of expert system applications. Configuration applications were pioneered by computer companies as a means of facilitating the manufacture of semi-custom minicomputers McDermott  The technique has found its way into use in many different industries, for example, modular home building, manufacturing, and other problems involving complex engineering design and manufacturing. Financial Decision Making The financial services industry has been a vigorous user of expert system techniques. Advisory programs have been created to assist bankers in determining whether to make loans to businesses and individuals. Insurance companies have used expert systems to assess the risk presented by the customer and to determine a price for the insurance. A typical application in the financial markets is in foreign exchange trading. Knowledge Publishing This is a relatively new, but also potentially explosive area. The two most widely distributed expert systems in the world are in this category. The first is an advisor which counsels a user on appropriate grammatical usage in a text. The second is a tax advisor that accompanies a tax preparation program and advises the user on tax strategy, tactics, and individual tax policy. Process Monitoring and Control Systems falling in this class analyze real-time data from physical devices with the goal of noticing anomalies, predicting trends, and controlling for both optimality and failure correction. Examples of real-time systems that actively monitor processes can be found in the steel making and oil refining industries. Design and Manufacturing These systems assist in the design of physical devices and processes, ranging from high-level conceptual design of abstract entities all the way to factory floor configuration of manufacturing processes.

# EXPERT SYSTEM TECHNOLOGY pdf

## 5: Expert system - Wikipedia

*In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if-then rules rather than through conventional procedural code.*

Expert Systems and Applied Artificial Intelligence  The field of artificial intelligence AI is concerned with methods of developing systems that display aspects of intelligent behaviour. These systems are designed to imitate the human capabilities of thinking and sensing. Symbolic Processing In AI applications, computers process symbols rather than numbers or letters. AI applications process strings of characters that represent real-world entities or concepts. Symbols can be arranged in structures such as lists, hierarchies, or networks. These structures show how symbols relate to each other. Nonalgorithmic Processing Computer programs outside the AI domain are programmed algorithms; that is, fully specified step-by-step procedures that define a solution to the problem. The actions of a knowledge-based AI system depend to a far greater degree on the situation where it is used. The Field of AI Artificial intelligence is a science and technology based on disciplines such as computer science, biology, psychology, linguistics, mathematics, and engineering. The goal of AI is to develop computers that can think, see, hear, walk, talk and feel. A major thrust of AI is the development of computer functions normally associated with human intelligence, such as reasoning, learning, and problem solving. General problem-solving methods AI established as research field. Knowledge-based expert systems Result: Transaction processing and decision support systems using AI. Resembling the interconnected neuronal structures in the human brain Intelligent agents Result: Software that performs assigned tasks on the users behalf  General View The most important applied area of AI is the field of expert systems. An expert system ES is a knowledge-based system that employs knowledge about its application domain and uses an inferencing reason procedure to solve problems that would otherwise require human competence or expertise. It is important to stress to students that expert systems are assistants to decision makers and not substitutes for them. Expert systems do not have human capabilities. They use a knowledge base of a particular domain and bring that knowledge to bear on the facts of the particular situation at hand. The knowledge base of an ES also contains heuristic knowledge - rules of thumb used by human experts who work in the domain. These include areas such as high-risk credit decisions, advertising decision making, and manufacturing decisions. Application areas include classification, diagnosis, monitoring, process control, design, scheduling and planning, and generation of options. An ES is built in a process known as knowledge engineering, during which knowledge about the domain is acquired from human experts and other sources by knowledge engineers. The accumulation of knowledge in knowledge bases, from which conclusions are to be drawn by the inference engine, is the hallmark of an expert system. Knowledge Representation and the Knowledge Base The knowledge base of an ES contains both factual and heuristic knowledge. Knowledge representation is the method used to organize the knowledge in the knowledge base. Knowledge bases must represent notions as actions to be taken under circumstances, causality, time, dependencies, goals, and other higher-level concepts. Several methods of knowledge representation can be drawn upon. Two of these methods include: Frame-based systems - are employed for building very powerful ESs. A frame specifies the attributes of a complex object and frames for various object types have specified relationships. Production rules - are the most common method of knowledge representation used in business. Rule-based expert systems are expert systems in which the knowledge is represented by production rules. A production rule, or simply a rule, consists of an IF part a condition or premise and a THEN part an action or conclusion. The explanation facility explains how the system arrived at the recommendation. Depending on the tool used to implement the expert system, the explanation may be either in a natural language or simply a listing of rule numbers. Inference Engine [Figure  Combines the facts of a specific case with the knowledge contained in the knowledge base to come up with a recommendation. In a rule-based expert system, the inference engine controls the order in which production rules are applied Afired and resolves conflicts if more than one rule is applicable at a given time. This is what Areasoning amounts to in rule-based systems. Directs the user

interface to query the user for any information it needs for further inferencing. The facts of the given case are entered into the working memory, which acts as a blackboard, accumulating the knowledge about the case at hand. The inference engine repeatedly applies the rules to the working memory, adding new information obtained from the rules conclusions to it, until a goal state is produced or confirmed. Inferencing engines for rule-based systems generally work by either forward or backward chaining of rules. Forward chaining - is a data-driven strategy. The inferencing process moves from the facts of the case to a goal conclusion. The strategy is thus driven by the facts available in the working memory and by the premises that can be satisfied. The inference engine attempts to match the condition IF part of each rule in the knowledge base with the facts currently available in the working memory. If several rules match, a conflict resolution procedure is invoked; for example, the lowest-numbered rule that adds new information to the working memory is fired. The conclusion of the firing rule is added to the working memory. Forward-chaining systems are commonly used to solve more open-ended problems of a design or planning nature, such as, for example, establishing the configuration of a complex product. Backward chaining - the inference engine attempts to match the assumed hypothesized conclusion - the goal or subgoal state - with the conclusion THEN part of the rule. If such a rule is found, its premise becomes the new subgoal. In an ES with few possible goal states, this is a good strategy to pursue. If a hypothesized goal state cannot be supported by the premises, the system will attempt to prove another goal state. Thus, possible conclusions are review until a goal state that can be supported by the premises is encountered. Backward chaining is best suited for applications in which the possible conclusions are limited in number and well defined. Classification or diagnosis type systems, in which each of several possible conclusions can be checked to see if it is supported by the data, are typical applications. Uncertainty and Fuzzy Logic Fuzzy logic is a method of reasoning that resembles human reasoning since it allows for approximate values and inferences and incomplete or ambiguous data fuzzy data. Fuzzy logic is a method of choice for handling uncertainty in some expert systems. Expert systems with fuzzy-logic capabilities thus allow for more flexible and creative handling of problems. These systems are used, for example, to control manufacturing processes. Two important things to keep in mind when selecting ES tools include: The tool selected for the project has to match the capability and sophistication of the projected ES, in particular, the need to integrate it with other subsystems such as databases and other components of a larger information system. The tool also has to match the qualifications of the project team. Expert systems technologies include: Expert system shells - are the most common vehicle for the development of specific ESs. A shell is an expert system without a knowledge base. A shell furnishes the ES developer with the inference engine, user interface, and the explanation and knowledge acquisition facilities. Domain-specific shells are actually incomplete specific expert systems, which require much less effort in order to field an actual system. Expert system development environments - these systems expand the capabilities of shells in various directions. They run on engineering workstations, minicomputers, or mainframes; offer tight integration with large databases; and support the building of large expert systems. ESs are now rarely developed in a programming language. Expert - Successful ES systems depend on the experience and application of knowledge that the people can bring to it during its development. Large systems generally require multiple experts. Knowledge engineer - The knowledge engineer has a dual task. This person should be able to elicit knowledge from the expert, gradually gaining an understanding of an area of expertise. Intelligence, tact, empathy, and proficiency in specific techniques of knowledge acquisition are all required of a knowledge engineer. Knowledge-acquisition techniques include conducting interviews with varying degrees of structure, protocol analysis, observation of experts at work, and analysis of cases. On the other hand, the knowledge engineer must also select a tool appropriate for the project and use it to represent the knowledge with the application of the knowledge acquisition facility. User - A system developed by an end user with a simple shell, is built rather quickly an inexpensively. Larger systems are built in an organized development effort. A prototype-oriented iterative development strategy is commonly used. ESs lends themselves particularly well to prototyping. Problem Identification and Feasibility Analysis: The needed degree of integration with other subsystems and databases is established - concepts that best represent the domain knowledge are worked out - the best way to represent the knowledge and to perform inferencing should be established with sample cases 3. Testing and Refinement

of Prototype: End users test the prototypes of the ES. Complete and Field the ES: Benefits and Limitations Expert systems offer both tangible and important intangible benefits to owner companies. These benefits should be weighted against the development and exploitation costs of an ES, which are high for large, organizationally important ESs. But these systems can dramatically reduce the amount of work the individual must do to solve a problem, and they do leave people with the creative and innovative aspects of problem solving. Some of the possible organizational benefits of expert systems are: An Es can complete its part of the tasks much faster than a human expert. The error rate of successful systems is low, sometimes much lower than the human error rate for the same task. ESs make consistent recommendations 4.

# EXPERT SYSTEM TECHNOLOGY pdf

## 6: EXPERT SYSTEM â€“ NATIONAL INFORMATICS EXPERT SYSTEM TECHNOLOGY By J. Ashraf | be

*Section - A: Present & Future Perspectives EXPERT SYSTEM - NATIONAL INFORMATICS EXPERT SYSTEM TECHNOLOGY Prof. S.J. Ashraf* 1. INTRODUCTION It has been said that non experts outnumber experts in any particular field by a ratio of 1 ; perhaps, this is truer of the developing countries.*

It would match R1 and assert Mortal Socrates into the knowledge base. Backward chaining is a bit less straight forward. In backward chaining the system looks at possible conclusions and works backward to see if they might be true. So if the system was trying to determine if Mortal Socrates is true it would find R1 and query the knowledge base to see if Man Socrates is true. One of the early innovations of expert systems shells was to integrate inference engines with a user interface. This could be especially powerful with backward chaining. So in this example, it could use R1 to ask the user if Socrates was a Man and then use that new information accordingly. The use of rules to explicitly represent knowledge also enabled explanation abilities. In the simple example above if the system had used R1 to assert that Socrates was Mortal and a user wished to understand why Socrates was mortal they could query the system and the system would look back at the rules which fired to cause the assertion and present those rules to the user as an explanation. In English if the user asked "Why is Socrates Mortal? A significant area for research was the generation of explanations from the knowledge base in natural English rather than simply by showing the more formal but less intuitive rules. These systems record the dependencies in a knowledge-base so that when facts are altered, dependent knowledge can be altered accordingly. For example, if the system learns that Socrates is no longer known to be a man it will revoke the assertion that Socrates is mortal. In this, the knowledge base can be divided up into many possible views, a. This allows the inference engine to explore multiple possibilities in parallel. For example, the system may want to explore the consequences of both assertions, what will be true if Socrates is a Man and what will be true if he is not? One of the first extensions of simply using rules to represent knowledge was also to associate a probability with each rule. So, not to assert that Socrates is mortal, but to assert Socrates may be mortal with some probability value. Simple probabilities were extended in some systems with sophisticated mechanisms for uncertain reasoning and combination of probabilities. With the addition of object classes to the knowledge base, a new type of reasoning was possible. Along with reasoning simply about object values, the system could also reason about object structures. In this simple example, Man can represent an object class and R1 can be redefined as a rule that defines the class of all men. These types of special purpose inference engines are termed classifiers. Although they were not highly used in expert systems, classifiers are very powerful for unstructured volatile domains, and are a key technology for the Internet and the emerging Semantic Web. With an expert system the goal was to specify the rules in a format that was intuitive and easily understood, reviewed, and even edited by domain experts rather than IT experts. The benefits of this explicit knowledge representation were rapid development and ease of maintenance. Ease of maintenance is the most obvious benefit. This was achieved in two ways. First, by removing the need to write conventional code, many of the normal problems that can be caused by even small changes to a system could be avoided with expert systems. Essentially, the logical flow of the program at least at the highest level was simply a given for the system, simply invoke the inference engine. This also was a reason for the second benefit: With an expert system shell it was possible to enter a few rules and have a prototype developed in days rather than the months or year typically associated with complex IT projects. A claim for expert system shells that was often made was that they removed the need for trained programmers and that experts could develop systems themselves. In reality, this was seldom if ever true. While the rules for an expert system were more comprehensible than typical computer code, they still had a formal syntax where a misplaced comma or other character could cause havoc as with any other computer language. Also, as expert systems moved from prototypes in the lab to deployment in the business world, issues of integration and maintenance became far more critical. Inevitably demands to integrate with, and take advantage of, large legacy databases and systems arose. To accomplish this, integration required the same skills as any other type of system. Obtaining the time of domain experts for any software application is always difficult, but for expert systems it was especially

difficult because the experts were by definition highly valued and in constant demand by the organization. As a result of this problem, a great deal of research in the later years of expert systems was focused on tools for knowledge acquisition, to help automate the process of designing, debugging, and maintaining rules defined by experts. However, when looking at the life-cycle of expert systems in actual use, other problems â€" essentially the same problems as those of any other large system â€" seem at least as critical as knowledge acquisition: This provided a powerful development environment, but with the drawback that it was virtually impossible to match the efficiency of the fastest compiled languages such as C. System and database integration were difficult for early expert systems because the tools were mostly in languages and platforms that were neither familiar to nor welcome in most corporate IT environments â€" programming languages such as Lisp and Prolog, and hardware platforms such as Lisp machines and personal computers. As a result, much effort in the later stages of expert system tool development was focused on integrating with legacy environments such as COBOL and large database systems, and on porting to more standard platforms. These issues were resolved mainly by the client-server paradigm shift, as PCs were gradually accepted in the IT environment as a legitimate platform for serious business system development and as affordable minicomputer servers provided the processing power needed for AI applications. The example applications were not in the original Hayes-Roth table, and some of them arose well afterward. Any application that is not footnoted is described in the Hayes-Roth book.

## 7: Expert Systems with Applications - Journal - Elsevier

*The technology may obtain expert systems that are entirely described in a graphical manner without any line of textual code. This chapter describes what is required of an expert system development environment and shows why the visual approach is well suited to meeting those requirements.*

This type of reasoning can be imitated by using numeric values called confidences. For example, if it is known that Fritz is green, it might be concluded with 0. These certainty factor CF numbers quantify uncertainty in the degree to which the available evidence supports a hypothesis. They represent a degree of confirmation and are not probabilities in a Bayesian sense. It can be mapped to a probability update, although degrees of confirmation are not expected to obey the laws of probability. Chaining Two methods of reasoning when using inference rules are backward chaining and forward chaining. Forward chaining starts with the data available and uses the inference rules to conclude more data until a desired goal is reached. An inference engine using forward chaining searches the inference rules until it finds one in which the if clause is known to be true. It then concludes the then clause and adds this information to its data. It would continue to do this until a goal is reached. Because the data available determines which inference rules are used, this method is also called data driven. Backward chaining starts with a list of goals and works backwards to see if there is data which will allow it to conclude any of these goals. An inference engine using backward chaining would search the inference rules until it finds one which has a then clause that matches a desired goal. If the if clause of that inference rule is not known to be true, then it is added to the list of goals. For example, suppose a rule base contains If Fritz is green then Fritz is a frog. If Fritz is a frog then Fritz hops. Suppose a goal is to conclude that Fritz hops. The rule base would be searched and rule 2 would be selected because its conclusion the then clause matches the goal. It is not known that Fritz is a frog, so this "if" statement is added to the goal list. The rule base is again searched and this time rule 1 is selected because its then clause matches the new goal just added to the list. This time, the if clause Fritz is green is known to be true and the goal that Fritz hops is concluded. Because the list of goals determines which rules are selected and used, this method is called goal driven. Expert system architecture The following general points about expert systems and their architecture have been illustrated. The sequence of steps taken to reach a conclusion is dynamically synthesized with each new case. It is not explicitly programmed when the system is built. Expert systems can process multiple values for any problem parameter. This permits more than one line of reasoning to be pursued and the results of incomplete not fully determined reasoning to be presented. Problem solving is accomplished by applying specific knowledge rather than specific technique. This is a key idea in expert systems technology. It reflects the belief that human experts do not process their knowledge differently from others, but they do possess different knowledge. With this philosophy , when one finds that their expert system does not produce the desired results, work begins to expand the knowledge base, not to re-program the procedures. There are various expert systems in which a rulebase and an inference engine cooperate to simulate the reasoning process that a human expert pursues in analyzing a problem and arriving at a conclusion. In these systems, in order to simulate the human reasoning process, a vast amount of knowledge needed to be stored in the knowledge base. Generally, the knowledge base of such an expert system consisted of a relatively large number of "if then" type of statements that were interrelated in a manner that, in theory at least, resembled the sequence of mental steps that were involved in the human reasoning process. Because of the need for large storage capacities and related programs to store the rulebase, most expert systems have, in the past, been run only on large information handling systems. Recently, the storage capacity of personal computers has increased to a point where it is becoming possible to consider running some types of simple expert systems on personal computers. In some applications of expert systems, the nature of the application and the amount of stored information necessary to simulate the human reasoning process for that application is just too vast to store in the active memory of a computer. In other applications of expert systems, the nature of the application is such that not all of the information is always needed in the reasoning process. An example of this latter type application would be the use of an expert system to diagnose a data processing system comprising many

separate components, some of which are optional. When that type of expert system employs a single integrated rulebase to diagnose the minimum system configuration of the data processing system, much of the rulebase is not required since many of the components which are optional units of the system will not be present in the system. Nevertheless, earlier expert systems require the entire rulebase to be stored since all the rules were, in effect, chained or linked together by the structure of the rulebase. When the rulebase is segmented, preferably into contextual segments or units, it is then possible to eliminate portions of the Rulebase containing data or knowledge that is not needed in a particular application. The segmenting of the rulebase also allows the expert system to be run with systems or on systems having much smaller memory capacities than was possible with earlier arrangements since each segment of the rulebase can be paged into and out of the system as needed. The segmenting of the rulebase into contextual segments requires that the expert system manage various intersegment relationships as segments are paged into and out of memory during execution of the program. Since the system permits a rulebase segment to be called and executed at any time during the processing of the first rulebase, provision must be made to store the data that has been accumulated up to that point so that at some time later in the process, when the system returns to the first segment, it can proceed from the last point or rule node that was processed. Also, provision must be made so that data that has been collected by the system up to that point can be passed to the second segment of the rulebase after it has been paged into the system and data collected during the processing of the second segment can be passed to the first segment when the system returns to complete processing that segment. The user interface and the procedure interface are two important functions in the information collection process. End user The end-user usually sees an expert system through an interactive dialog, an example of which follows: Do you know which restaurant you want to go to? Is there any kind of food you would particularly like? Do you like spicy food? Do you usually drink wine with meals? When you drink wine, is it French wine? Yes As can be seen from this dialog , the system is leading the user through a set of questions , the purpose of which is to determine a suitable set of restaurants to recommend. This dialog begins with the system asking if the user already knows the restaurant choice a common feature of expert systems and immediately illustrates a characteristic of expert systems; users may choose not to respond to any question. In expert systems, dialogs are not pre-planned. There is no fixed control structure. Dialogs are synthesized from the current information and the contents of the knowledge base. Explanation system Another major distinction between expert systems and traditional systems is illustrated by the following answer given by the system when the user answers a question with another question, "Why", as occurred in the above example. I am trying to determine the type of restaurant to suggest. So far Chinese is not a likely choice. It is possible that French is a likely choice. I know that if the diner is a wine drinker, and the preferred wine is French, then there is strong evidence that the restaurant choice should include French. It is very difficult to implement a general explanation system answering questions like "Why" and "How" in a traditional computer program. An expert system can generate an explanation by retracing the steps of its reasoning. The response of the expert system to the question WHY is an exposure of the underlying knowledge structure. It is a rule; a set of antecedent conditions which, if true, allow the assertion of a consequent. The rule references values, and tests them against various constraints or asserts constraints onto them. This, in fact, is a significant part of the knowledge structure. There are values, which may be associated with some organizing entity. For example, the individual diner is an entity with various attributes values including whether they drink wine and the kind of wine. There are also rules, which associate the currently known values of some attributes with assertions that can be made about other attributes. It is the orderly processing of these rules that dictates the dialog itself. Expert systems versus problem-solving systems The principal distinction between expert systems and traditional problem solving programs is the way in which the problem related expertise is coded. In traditional applications, problem expertise is encoded in both program and data structures. In the expert system approach all of the problem related expertise is encoded in data structures only; no problem-specific information is encoded in the program structure. This organization has several benefits. An example may help contrast the traditional problem solving program with the expert system approach. The example is the problem of tax advice. In the expert system approach, the information about taxpayers and tax computations is again found in data

structures, but now the knowledge describing the relationships between them is encoded in data structures as well. The programs of an expert system are independent of the problem domain taxes and serve to process the data structures without regard to the nature of the problem area they describe. For example, there are programs to acquire the described data values through user interaction , programs to represent and process special organizations of description, and programs to process the declarations that represent semantic relationships within the problem domain and an algorithm to control the processing sequence and focus. The general architecture of an expert system involves two principal components: Individuals involved with expert systems There are generally three individuals having an interaction with expert systems. Primary among these is the end-user ; the individual who uses the system for its problem solving assistance. In the building and maintenance of the system there are two other roles: Usually, the knowledge engineer will represent the problem solving activity in the form of rules which is referred to as a rule-based expert system. When these rules are created from the domain expertise, the knowledge base stores the rules of the expert system. Inference rule An understanding of the " inference rule " concept is important to understand expert systems. An inference rule is a statement that has two parts, an if clause and a then clause. This rule is what gives expert systems the ability to find solutions to diagnostic and prescriptive problems. An example of an inference rule is: If the restaurant choice includes French, and the occasion is romantic, Then the restaurant choice is definitely Paul Bocuse. They are entered as separate rules and it is the inference engine that uses them together to draw conclusions. Because each rule is a unit, rules may be deleted or added without affecting other rules though it should affect which conclusions are reached. One advantage of inference rules over traditional programming is that inference rules use reasoning which more closely resemble human reasoning. Thus, when a conclusion is drawn, it is possible to understand how this conclusion was reached. Furthermore, because the expert system uses knowledge in a form similar to the expert , it may be easier to retrieve this information from the expert. Procedure node interface The function of the procedure node interface is to receive information from the procedures coordinator and create the appropriate procedure call.

# EXPERT SYSTEM TECHNOLOGY pdf

## 8: Expert Systems and Applied Artificial Intelligence

*These caveats notwithstanding, expert systems have demonstrated their value to the business community, and managers should assume responsibility for using ES technology thoughtfully.*

Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge. The data is collection of facts. The information is organized as data and facts about the task domain. Data, information, and past experience combined together are termed as knowledge. Components of Knowledge Base The knowledge base of an ES is a store of both, factual and heuristic knowledge. Knowledge representation It is the method used to organize and formalize the knowledge in the knowledge base. Knowledge Acquisition The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base. The knowledge base is formed by readings from various experts, scholars, and the Knowledge Engineers. The knowledge engineer is a person with the qualities of empathy, quick learning, and case analyzing skills. He acquires information from subject expert by recording, interviewing, and observing him at work, etc. The knowledge engineer also monitors the development of the ES. Inference Engine Use of efficient procedures and rules by the Inference Engine is essential in deducting a correct, flawless solution. In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution. Adds new knowledge into the knowledge base if required. Resolves rules conflict when multiple rules are applicable to a particular case. It considers all the facts and rules, and sorts them before concluding to a solution. This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence. It explains how the ES has arrived at a particular recommendation. Verbal narrations in natural language. Listing of rule numbers displayed on the screen. The user interface makes it easy to trace the credibility of the deductions. It should make efficient use of user input. Expert Systems Limitations No technology can offer easy and complete solution. Large systems are costly, require significant development time, and computer resources.

## 9: Artificial Intelligence Expert Systems

*One example of an expert system is an artificial intelligence system that emulates an auto mechanic's knowledge in diagnosing automobile problems. This hypothetical expert system would likely be the result of engineering using an actual mechanic's knowledge base. Expert systems are designed by.*

# EXPERT SYSTEM TECHNOLOGY pdf

*Grandfathers Private Zoo A treatise on the law of awards Insiders and favorites Employee handbooks mrhandyman filetype Fortran learn for dummies Please Walk Beside Me A Certain Curve of Horn Writing With Reason Presidency, its duties, its powers, its opportunities and its limitations Developing protocols for obstetric emergencies Equal opportunity in federal construction Insight Compact Guide Normandy Booknotes on American Character Postcolonial international relations Handbook of Crisis and Emergency Management (Public Administration and Public Policy) Juvenile justice act 2015 Prevailing Prayer Satellite information systems Filetype japanese for busy people ii The flaying season Toward a suitably complex framework of analysis Frogments from the Frag Pool Health Care And Poor Relief In 18th And 19th Century Southern Europe (The History of Medicine in Context) Cross Fertilization: The Human Spirit As Place (Contemporary Anthology Series : No. 3) Fifty uncommon birds of the upper Midwest Kia picanto manual book Magnificent Obsessions Poetry Of Thomas Moore Rotating machinery Profile of the typical developing middle school student with autism/AS Characteristics of compassion Toys r us christmas book The Life of John Marshall (Volume II) Opera : a special case? Gods purpose for every woman Wavelet methods for elliptic partial differential equations The New Fish Cooking Encyclopedia My grandmother Millard and General Bedford Forrest and the Battle of Harrykin Creek Families in the New Millennium Planet Earth: 25 Environmental Projects You Can Build Yourself*