Chapter 1 : Expert Systems/Fuzzy Logic - Wikibooks, open books for an open world

Expert Systems With Applications is a refereed international journal whose focus is on exchanging information relating to expert and intelligent systems applied in industry, government, and universities worldwide. The thrust of the journal is to publish papers dealing with the design, development, testing.

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.

Chapter 2 : Expert Systems with Applications - Journal - Elsevier

Enter your mobile number or email address below and we'll send you a link to download the free Kindle App. Then you can start reading Kindle books on your smartphone, tablet, or computer - no Kindle device required.

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

DOWNLOAD PDF EXPERT SYSTEMS, THEORY AND APPLICATIONS

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.

Chapter 3 : Fuzzy logic - Wikipedia

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

This article was written on 03 Jan , and is filled under Volume 5 No 4. I can sense their minds searching for mysterious elusive theories that they conclude that they must have missed. Only a few realize that many theories that they are already familiar with have great applicability to nursing informatics. One such theory is the time honoured Novice to Expert theory. The Novice to Expert Theory, a construct theory first proposed by Hubert and Stuart Dreyfus as the Dreyfus Model of Skill Acquisition, and later applied and modified to nursing by Patricia Benner provides a very useful and important theory that clearly applies to nursing informatics. The Dreyfus brothers developed the model while working with scholars interested in comparing artificial intelligence development and expert computer system programming to the human mind and the development of expertise. Within the field of nursing informatics, this theory can be applied to: The currently accepted five levels of development within the Novice to Expert theoretical model are illustrated in the image above, as presented by Benner They start from the bottom rung at the Novice level and move upward through Advanced Beginner, Competent, Proficient, and Expert levels. Dreyfus and Dreyfus initially proposed the stages of: Novice, Competent, Proficient, Expertise and Mastery. In both configurations, each level builds on the level before it as the learner advances from a neophyte level then gains knowledge, skills, perceptions, intuition, wisdom and most important of all, experience in their given field of practice. Distinguishing Traits Both Dreyfus and Dreyfus and Benner estimated that it takes approximately five years to move through the five stages from novice to expert but also elaborated that not all novices become experts. Deliberate practice is a trait shown by people who use a personal, goal-oriented approach to skill and knowledge development â€" they devote themselves to engage in progressively higher, and ultimately expert performance. This requires years of sustained effort to continually improve the quality of their practice and performance within the skill â€" in this case, in nursing informatics skills. They feel personal satisfaction in confronting challenges to achieve a high standard of excellence within their field. They are not content to acquire merely functional and rudimentary skill levels â€" they want to shine, and join the ranks of the experts in the field. This is a quality often seen in Super Users and Champions within the nursing informatics arena. To move to this level, many different perspectives must be digested and the zone of comfort can become threatening. Many people do not like to stand out from the rest, so do not risk the possibility of being perceived as different or peculiar â€" nor do they want to be regarded as thinking that they excel above their peers. Yet, the true expert must take this risk and continue to move up the ladder of skill and knowledge acquisition despite potential conflict within the nursing workplace. Some common themes are evident as a person successfully progresses through the novice to expert levels: As progression occurs, the person tends to move away from relying on rules and explicit knowledge to learning to trust and follow their intuition and pattern matching. Better cognitive filtering occurs, where problems are no longer a huge confusing collection of data but instead become a complete and unique whole where some bits are much more relevant than others. The person also moves from being a detached observer of a problem to an involved part of the system itself, accepting responsibility for results, not just for carrying out tasks. Novice to Expert Levels Each of the five levels of skill acquisition has distinguishing behaviours and traits Frisoli, The novice is then given rules for determining an action on the basis of these features. To improve, the novice needs monitoring, either by self-observation or instructional feedback. Advanced Beginner An advanced beginner is still dependent on rules, but as s he gains more experience with real life situations, s he begins to notice additional aspects that can be applied to related conditions. Proficient The fourth stage is called fluency and is characterized by the progress of the learner from the step-by-step analysis and solving of the situation to the holistic perception of the entirety of the situation. The proficient hospital information system learner would know how to interpret data from all departmental information and provide guidance to other disciplinary members as needed. After a great deal of experience actually using a system in everyday situations, the expert nurse discovers that without his consciously using any rules, situations simply elicit from him or her appropriate responses. The proficient performer, immersed in the world of his skillful activity, sees what needs to be done, and decides how to do it. Change leaders and project managers would do well to consider the Novice to Expert theory as they plan implementation initiatives and other informatics training opportunities. Learners need particular types of guidance, depending on the level they are currently at. Novices need much different instruction and support than advanced beginners need, and so on. The success of your initiative ultimately depends on the end-users. Why not apply this simple yet effective theory to your nursing informatics practice? From novice to expert: Excellence and power in clinical nursing. A Five-Stage Model of the mental activities involved in direct skill acquisition. Operations Research Center Report. University of California, Berkeley. Adult Learning and Technology. Theory applied to informatics â€” Novice to Expert. Canadian Journal of Nursing Informatics, 5 4 , Editorial.

Chapter 4 : Recent Expert Systems with Applications Articles - Elsevier

Recently published articles from Expert Systems with Applications.

There are also other operators, more linguistic in nature, called hedges that can be applied. These are generally adverbs such as very, or somewhat, which modify the meaning of a set using a mathematical formula. In the paper, [9] a criterion has been formulated to recognize whether a given choice table defines a fuzzy logic function and a simple algorithm of fuzzy logic function synthesis has been proposed based on introduced concepts of constituents of minimum and maximum. A fuzzy logic function represents a disjunction of constituents of minimum, where a constituent of minimum is a conjunction of variables of the current area greater than or equal to the function value in this area to the right of the function value in the inequality, including the function value. Defuzzification The goal is to get a continuous variable from fuzzy truth values. Since, however, all output truth values are computed independently, in most cases they do not represent such a set of numbers. A common algorithm is For each truth value, cut the membership function at this value Combine the resulting curves using the OR operator Find the center-of-weight of the area under the curve The x position of this center is then the final output. Forming a consensus of inputs and fuzzy rules[edit] Since the fuzzy system output is a consensus of all of the inputs and all of the rules, fuzzy logic systems can be well behaved when input values are not available or are not trustworthy. Weightings can be optionally added to each rule in the rulebase and weightings can be used to regulate the degree to which a rule affects the output values. These rule weightings can be based upon the priority, reliability or consistency of each rule. These rule weightings may be static or can be changed dynamically, even based upon the output from other rules. Early applications[edit] Many of the early successful applications of fuzzy logic were implemented in Japan. The first notable application was on the subway train in Sendai , in which fuzzy logic was able to improve the economy, comfort, and precision of the ride. Propositional fuzzy logics[edit] The most important propositional fuzzy logics are: Monoidal t-norm-based propositional fuzzy logic MTL is an axiomatization of logic where conjunction is defined by a left continuous t-norm and implication is defined as the residuum of the t-norm. Its models correspond to MTL-algebras that are pre-linear commutative bounded integral residuated lattices. Basic propositional fuzzy logic BL is an extension of MTL logic where conjunction is defined by a continuous t-norm, and implication is also defined as the residuum of the t-norm. Its models correspond to BL-algebras. It has the axioms of basic fuzzy logic plus an axiom of double negation, and its models correspond to MV-algebras. It has the axioms of BL plus an axiom of idempotence of conjunction, and its models are called G-algebras. Product fuzzy logic is the extension of basic fuzzy logic BL where conjunction is product t-norm. It has the axioms of BL plus another axiom for cancellativity of conjunction, and its models are called product algebras. This means that each formula has an evaluation. Predicate fuzzy logics[edit] These extend the above-mentioned fuzzy logics by adding universal and existential quantifiers in a manner similar to the way that predicate logic is created from propositional logic. The semantics of the universal resp. Decidability issues for fuzzy logic[edit] The notions of a "decidable subset" and " recursively enumerable subset" are basic ones for classical mathematics and classical logic. Thus the question of a suitable extension of them to fuzzy set theory is a crucial one. A first proposal in such a direction was made by E. Santos by the notions of fuzzy Turing machine , Markov normal fuzzy algorithm and fuzzy program see Santos Gerla argued that the proposed definitions are rather questionable. For example, in [12] one shows that the fuzzy Turing machines are not adequate for fuzzy language theory since there are natural fuzzy languages intuitively computable that cannot be recognized by a fuzzy Turing Machine. Then, they proposed the following definitions. Then a fuzzy subset s: We say that s is decidable if both s and its complement â€"s are recursively enumerable. An extension of such a theory to the general case of the L-subsets is possible see Gerla The proposed definitions are well related with fuzzy logic. Indeed, the following theorem holds true provided that the deduction apparatus of the considered fuzzy logic satisfies some obvious effectiveness property. Any "axiomatizable" fuzzy theory is recursively enumerable. In particular, the fuzzy set of logically true formulas is recursively enumerable in spite of the fact that the crisp set of valid formulas is not recursively enumerable, in general. Moreover, any axiomatizable and complete theory is decidable. It is an open question to give supports for a "Church thesis" for fuzzy mathematics , the proposed notion of recursive enumerability for fuzzy subsets is the adequate one. In order to solve this, an extension of the notions of fuzzy grammar and fuzzy Turing machine are necessary. Fuzzy databases[edit] Once fuzzy relations are defined, it is possible to develop fuzzy relational databases. Fuzzy querying languages have been defined, such as the SQLf by P. These languages define some structures in order to include fuzzy aspects in the SQL statements, like fuzzy conditions, fuzzy comparators, fuzzy constants, fuzzy constraints, fuzzy thresholds, linguistic labels etc. Comparison to probability[edit] Fuzzy logic and probability address different forms of uncertainty. While both fuzzy logic and probability theory can represent degrees of certain kinds of subjective belief, fuzzy set theory uses the concept of fuzzy set membership, i. The concept of fuzzy sets was developed in the mid-twentieth century at Berkeley [13] as a response to the lacking of probability theory for jointly modelling uncertainty and vagueness. Probability [15] that probability theory is a subtheory of fuzzy logic, as questions of degrees of belief in mutually-exclusive set membership in probability theory can be represented as certain cases of non-mutually-exclusive graded membership in fuzzy theory. Zadeh argues that fuzzy logic is different in character from probability, and is not a replacement for it. He fuzzified probability to fuzzy probability and also generalized it to possibility theory. Relation to ecorithms[edit] Computational theorist Leslie Valiant uses the term ecorithms to describe how many less exact systems and techniques like fuzzy logic and "less robust" logic can be applied to learning algorithms. Valiant essentially redefines machine learning as evolutionary. In general use, ecorithms are algorithms that learn from their more complex environments hence eco- to generalize, approximate and simplify solution logic. Like fuzzy logic, they are methods used to overcome continuous variables or systems too complex to completely enumerate or understand discretely or exactly. Compensatory fuzzy logic[edit] Compensatory fuzzy logic CFL is a branch of fuzzy logic with modified rules for conjunction and disjunction. When the truth value of one component of a conjunction or disjunction is increased or decreased, the other component is decreased or increased to compensate. This increase or decrease in truth value may be offset by the increase or decrease in another component. An offset may be blocked when certain thresholds are met. The conjunction is the geometric mean and its dual as conjunctive and disjunctive operators. FML allows modelling a fuzzy logic system in a human-readable and hardware independent way. The designers of fuzzy systems with FML have a unified and high-level methodology for describing interoperable fuzzy systems.

Chapter 5 : Artificial Intelligence Expert Systems

Fuzzy Expert Systems and Fuzzy Reasoning, with its expert presentation of both theory and application, is an excellent textbook for graduate and upper-level undergraduate students. In addition, this is essential reading for program designers and researchers in fuzzy sets, fuzzy logic, computer science, and artificial intelligence.

Even for individual issues, families, organizations, societies, and other systems are inherently involved and must be considered when attempting to understand and assist the individual. According to this theory, all systems are interrelated parts constituting an ordered whole and each subsystem influences other parts of the whole. There have been dozens of unofficial iterations of Systems Theory over the past few hundred years, applied to society, science, and many other areas. In the 20th century, multiple scientists, philosophers, and academics began to outline and define the structure of Systems Theory in their various disciplines; there are now systems theories for biology, cybernetics, and for social work. While the applications obviously vary depending on the discipline, all systems theories follow the concept of interrelated parts influencing one another as part of an ordered whole. Several prominent thinkers advanced Systems Theory in social work. Robert Merton is considered one of the founding fathers of modern sociology and significantly advanced Systems Theory through his progressive theories on functional analysis. She mentored and worked extensively with Alex Gitterman , who continues to develop Systems Theory through the Life Model. Case Study in Systems Theory The Pruett case study provides a concrete, real-world example of how Systems Theory is applied to understand how interrelated factors contribute to unhealthy actions. In this case, the client was engaging in risky behaviors drug abuse and unprotected sex and not attending school. She had not had contact with her father for five years, and some of her only memories of him involved him abusing drugs and arguing with her mother at home. In the Family Systems Theory, individuals must not be evaluated in isolation, but in the context of the family, as the family operates as a unit. Clearly, the client was missing one of the corners of the triangle and thus one of the pillars of healthy emotional development. Another concept is the family projection process, wherein the client suffers from the emotional dysfunction of the family unit. In this case, the client witnessed her father abusing drugs to self-medicate, so she imitated that behavior, thinking it might help her. The full complexities of this case go beyond the scope of this post, but it serves as an example of how a social worker must understand interrelated systems e. Issues Addressed by Systems Theory Systems Theory is used to develop a holistic view of individuals within an environment and is best applied to situations where several systems inextricably connect and influence one another. It can be employed in cases where contextual understandings of behavior will lead to the most appropriate practice interventions. The recommended interventions thus involved strengthening the missing part of her family unit, referring her to counseling services, and connecting her with academic support. There are many practice interventions available to social workers and their applications vary greatly depending on the context, but following are a few common interventions used as part of Systems Theory. Strengthen one part of the system to improve the whole. In the Pruett case, the social worker recommended finding a healthy father figure for the client, to strengthen the missing component of the family system. This often means referring clients to specialists, or connecting them with resources or organizations that can help their situation. In the Pruett case, this meant referral to a counselor and connection to an after school tutor. It allows social workers and clients to capture and organize the complexity of a system. A genogram is a graphic representation of a family tree, constructed with symbols that describe relationships and connections between an extended family. Social workers typically construct them along with clients in order to better understand relationships and identify patterns in the medical history. One of the most important functions of a social worker is helping clients navigate the various systems that affect their lives, which requires a deep understanding of how subsystems are interrelated and influence one another. This post provides an introduction to Systems Theory and some real life examples of how it is applied. It is just one of the many theoretical approaches that social workers will apply throughout their careers.

Chapter 6 : Theory applied to informatics â€“ Novice to Expert | Canadian Journal of Nursing Informatics

Bayesian Networks for Expert Systems, Theory and Practical Applications 3 however, data is often insufï¬•cient even for the quantitativ e part of the speciï¬•ca-.

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.

Chapter 7 : Expert Systems and Applied Artificial Intelligence

Fuzzy Expert Systems and Fuzzy Reasoning, with its expertpresentation of both theory and application, is an excellenttextbook for graduate and upper-level undergraduate students. Inaddition, this is essential reading for program designers andresearchers in fuzzy sets, fuzzy logic, computer science, andartificial intelligence.

If another person is 1. The crisp example differs deliberately from the fuzzy one. So, it is not so complex as being tall. Formal fuzzy logic[edit] In mathematical logic, there are several formal systems that model the above notions of "fuzzy logic"; most of them belong among so-called t-norm fuzzy logics. Note that they use a different set of operations than above mentioned Zadeh operators. Propositional fuzzy logics[edit] The most important propositional fuzzy logics are: Basic propositional fuzzy logic BL is an axiomatization of logic where conjunction is defined by a continuous t-norm, and implication is defined as the residuum of the t-norm. Its models correspond to BL-algebras. It has the axioms of basic logic plus an axiom of double negation so it is not intuitionistic logic , and its models correspond to MV-algebras. It has the axioms of basic logic plus an axiom of idempotence of conjunction, and its models are called G-algebras. Product fuzzy logic is a special case of basic fuzzy logic where conjunction is product t-norm. It has the axioms of basic logic plus another axiom, and its models are called product algebras. Monoidal t-norm logic MTL is a generalization of basic fuzzy logic BL where conjunction is realized by a left-continuous t-norm. Its models MTL-algebras are prelinear commutative bounded integral residuated lattices. Rational Pavelka logic is a generalization of multi-valued logic. All these logics encompass the traditional propositional logic whose models correspond to Boolean algebras. Predicate fuzzy logics[edit] These extend the above-mentioned fuzzy logics by adding universal and existential quantifiers in a manner similar to the way that predicate logic is created from propositional logic. The semantics of the universal resp. Effectiveness for fuzzy logics[edit] The notions of a "decidable subset" and "recursively enumerable subset" are basic ones for classical mathematics and classical logic. Then, the question of a suitable extension of such concepts to fuzzy set theory arises. A first proposal in such a direction was made by E. Santos by the notions of fuzzy Turing machine, Markov normal fuzzy algorithm and fuzzy program. An extension of such a theory to the general case of the L-subsets is proposed in a paper by G. The proposed definitions are well related with fuzzy logic. Indeed, the following theorem holds true provided that the deduction apparatus of the fuzzy logic satisfies some obvious effectiveness property. Any axiomatizable fuzzy theory is recursively enumerable. In particular, the fuzzy set of logically true formulas is recursively enumerable in spite of the fact that the crisp set of valid formulas is not recursively enumerable, in general. Moreover, any axiomatizable and complete theory is decidable. It is an open question to give supports for a Church thesis for fuzzy logic claiming that the proposed notion of recursive enumerability for fuzzy subsets is the adequate one. Bibliography[edit] Von Altrock, Constantin Fuzzy logic and NeuroFuzzy applications explained. Upper Saddle River, NJ: Archive for Mathematical Logic 41 7: The fuzzy systems handbook: IEEE Expert 9 4: Journal of Symbolic Logic 71 1: Metamathematics of fuzzy logic. Fuzzy Sets and Systems 3 8: Introduction to Applied Fuzzy Electronics. Fuzzy sets, uncertainty, and information. Fuzzy sets and fuzzy logic: Studia Logica 68 1: Algebraic foundations of many-valued reasoning. Mathematical principles of fuzzy logic. Journal of Symbolic Logic 27 2: Essentials of fuzzy modeling and control. Information and Control 12 2: Information and Control 8 3: Fuzzy Relational Data Bases. Fuzzy set theory and its applications.

Chapter 8 : IGCSE ICT - Expert Systems | IGCSE ICT

An expert system is a computer program that provides expert-level solutions to 'important problems and is: l heuristic --i.e., it reasons with judgmental knowledge as well as with formal knowledge of.

Chapter 9 : Expert system - Wikipedia

DOWNLOAD PDF EXPERT SYSTEMS, THEORY AND APPLICATIONS

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.