International Journal of Computer Theory and Engineering, Vol.3, No.1, February, 2011
1793-8201

Implementation of an Expert System as Spiritual Guru for Personality Development

Puja Shrivastava, Susanta Kumar Satpathy and Kapil Kumar Nagwanshi

Abstract—**Expert systems have been used in various fields of life like diagnosis and troubleshooting of devices and systems of all kinds, configuration of manufactured objects from subassemblies, planning and scheduling, financial decision making, knowledge publishing, process monitoring and control, design and manufacturing, agriculture, medicine, etc. Similarly these systems are also being used in field of training and education, and consultation because of its structured way of deriving knowledge and its explanation facility. This paper presents development of a web-based expert system as a Spiritual Guru or Guru of Life Ethics. This expert system is to cater those people who ask questions about the life and the world. It answers the questions as a spiritual guru, as a philosopher in addition to a scientist. Since the difference between in being a real scientist and being a real spiritual person is only that the scientist doesn't believe in the existence of soul and the spiritual person knows the soul or in the way of knowing it. The knowledge base of the proposed expert system will contain questions and their answers with reasoning. Inference engine will be a mechanism that will fetch keywords from working memory and match it with the questions stored in the knowledge base to answer the questions asked by the user. This paper presents how an expert system can be developed as Spiritual Guru to serve mankind, system development life cycle of expert system, general work-flow of Spiritual Guru, and designing of database.**

Index Terms—**Expert system, production rules, Spiritual Guru, self awareness, knowledge base.**

I. Introduction

It will be an apt solution for the environment where it seems that no individual is really serious about the life, life ethics and the existence of the world. In our daily lives understanding is almost nothing of the world and life. One can rarely think on it. One might think in mechanical way as the machinery that generates the sunlight that makes life possible, the gravity that glues objects to earth or the atoms of which living and nonliving beings are made and on whose stability the whole universe fundamentally depends [2].One can rarely think of questions like why nature is the way it is; where did the universe came from, and where is it going?, did the universe have a beginning, and if so, what happened before then?, what is the nature of time?, will it ever come to an end, and so forth [4]. In the same way one can never think of questions like what is life. Where the life has came from. Why death? What is next to death? What is the use of working hard while every one has to die finally? What is next to moksh? What is the purpose of life, etc.

Only a few peoples ask about the existence of life, purpose of life and such types of questions and they really need the answers, they never get satisfied with the present life style of society and world. They want the answer on birth, growth and death phases of life [5]. This is the destiny of every living being on this earth. In finding the solution of such questions the people reach to the religious people, who are rarely genuine people. There are more than 5000 religions in the world, each has its own initiator or Guru - founder of the religion, their own religious mythologies, rituals and the most dangerous thing is a thought "my religion is the original and the superior one, everyone should follow it, otherwise they should be killed", this thought has given birth to thousands of riots, battles and wars. At present terrorism (on the basis of religion) is the most critical problem and no solution is being seen.

This paper presents the development of an expert system that will be able to answer the questions mentioned above and telling the people about the importance of life ethics. This expert system will be able to cater people who are seeking a Guru of life, a Spiritual Guru. This system will be able to answer the questions in philosophical way and in a more scientific way too. This system will work as Spiritual guide. Spiritual guides are also known by the term guardian angels and other names. A true spirit guide is an evolved being who has agreed to support your spiritual evolution [6]. They respect people and their right to choose their own path. While spirit guides are always there for us, few people make the time to become aware of their presence, physical, mental, emotional and spiritual well-being.

The paper has been organized in the following manner; section II proposes design of an expert system, section III describes the characteristics of an expert system, section IV gives SDLC of proposed system followed by section describes the components of spiritual guru, finally section VI incorporates all the references been made for completion of this work

II. Design of An Expert System

Like a human expert, an expert system is expected to

- be specialist: know facts and procedural rules
- use heuristics: interpolate from known facts
- Justify its conclusions: to establish credibility and confidence.
- The user can ask: how do you know a particular fact? Why do you ask a particular question?

Computer Science & Engineering, Rungta College of Engineering & Technology, Bhilai, Chhatisgarh, India
pujashri@gmail.com, sks_sarita@yahoo.com, kapil14s@yahoo.co.uk

- be able to learn: be able to absorb new knowledge and apply it
- Estimate the reliability of its answer [9].
-

III. Characteristics Of Expert Systems

Expert systems use knowledge rather than data to provide the solution. The knowledge is encoded and maintained as an entity separate from the control program. Expert systems are capable of explaining how a particular conclusion was reached. Expert systems use symbolic representations for knowledge (rules, networks, or frames) and perform their inference through symbolic computations. Expert systems often reason with Meta Knowledge [7].

IV. SDLC of Proposed Systems

In its present form Spiritual Guru is a small scale expert system. The system development life cycle of Spiritual Guru contains following phases

1) Identification.
2) Conceptualization.
3) Formulation.
4) Implementation.
5) Testing.

Identification phase contains two parts-Project identification and selection and project initialization. Project Identification and Selection-this part identifies urgency of the project, how much a project will add cost and value to organization, expected budget, expected schedule and level of technical difficulties or constraints. Project Initialization-this part specifies problem or project definition, assessment of needs, generating alternative solution; verify approach to develop expert systems and management issues. In conceptualization phase knowledge engineer tries to conceptualize the expert system with respect to problem domain, such as what is the information required to develop the expert system and how it can be represented.

- To develop knowledge base of Spiritual Guru the domains is the philosophy of life, evolution of life and existence, explanations and proofs of both given by scientists and spiritual people.
- In Spiritual Guru Information will be represented in the form of questions and their answers and the reasons in the support of answers.

In formulation phase knowledge is gathered from the main sources of knowledge. Knowledge Engineer is to make a summary of requirements after performing the knowledge acquisition process. Implementation phase is the writing of programs to perform the desired task and linking of all programs to work as complete software. Testing phase is to check the software with raw and original data whether it is working without error.

V. Components of Spiritual Guru

Proposed ES will have basically five modules a knowledge base, inference engine, explanation subsystem, input/output interface and working memory. The knowledge base will have knowledge in the form of questions and their answers, such as question is "why I am living?" the answer of this question will be stored with it as "it is a divine wish". Explanation module of this system provides the reasoning, why the answer is this. Inference engine is the most important part of this system. It contains the implementation of logic that will take the keywords from the working memory. Workflow – Fig. 1 shows the different components of Spiritual Guru expert system, and how they work together to produce recommendations for the user.

In high level the algorithmic step can be described as follows:

A. Design Detail

Knowledge base design – the Knowledge Base (KB) of ES stores the extensive knowledge gathered from experts and books regarding the application in the form of rules. The knowledge is both factual and heuristic and stored in the form of production rules. Production rules are of the form of if-then-else rules.

The KB is divided into three major sections namely

1) Variable section
2) Rules section
3) Ask section

Variable section is a section used to declare all the variables used in the knowledge base. Rule section is used to define all the rules used to represent the knowledge. Ask section includes all the questions and options for the user to select and continue with the consultation assistance from the Spiritual Guru expert system.

Database design – gathered data can put in the form of questions and answers in the database of application. The database table of the ES shell have three fields namely – id, question, answer. The database can be accessed by entering the keyword related to the information searched. This leads to displaying of all the questions containing the keyword and selection of any one of the questions would provide for the displaying of the required information.

For designing database of Spiritual Guru Data Modelling is done with CLSQL. Before creating, query and manipulating CLSQL objects, it is needed to define data model of Spiritual Guru. Data modelling means telling to the relational database management system (RDBMS) the following:

- What elements of the data will be stored.
- How large each element can be.
- What kind of information each element can contain.
- What elements may be left blank.
- Which elements are constrained to a fixed range.
- Whether and how various tables are to be linked.

With SQL database one would do this by defining a set of relations, or tables, followed by a set of queries for joining the tables together in order to construct complex records. However, with CLSQL this is done by defining a set of CLOS(Common Lisp Object Store) classes, specifying how they will be turned into tables, and how they can be joined to one another via relations between their attributes. The SQL tables, as well as the queries for joining them together are created automatically, saving from dealing with some of the tedium of SQL.

Database of Spiritual Guru will basically contain three tables, one for the set of questions and answers, second for the user, and third for the administrator's information.
The Question-Answer table will have four elements of data – Question_Id, Question, Answer and Reasons.

The User table will contain information entered by the user to consult Spiritual Guru. This information will be the user's personnel information such as user's email-id, name, age, gender, relationship-status, profession, interests and something about the philosophy of life.

The Administrator table will contain administrator-id and password.

Simple SQL statements for the creation of these tables are

```
CREATE TABLE QUST_ANS
Question_Id   NOTNULL     number (10)
Question      NOTNULL     varchar2 (200)
Answer        NOTNULL     varchar2 (1000)
Reasons       NOTNULL     varchar2 (5000))
CREATE TABLE USER
 (Email_id     NOTNULL     varchar2 (50)
Name           NOTNULL     varchar2 (50)
Age            NOTNULL     number (3)
Gender         NOTNULL     varchar2 (1)
 Relationship_Status        varchar2 (10)
Profession                  varchar2 (50)
Interests                   varchar2 (100)
Philosophy                  varchar2 (500))
CREATE TABLE ADMINISTRATOR
(Admin-Id     NOTNULL   varchar2 (10)
Password varchar2 (10))
```

In CLSQL there will be three "view classes" (a fancy word for a class mapped into a database). They would be defined as follows:

```
 (clsql:def-view-class QUST_ANS ()
((Question_id
:db-kind :key
:db-constraints :not-null
:type integer
:initarg :Question_id)
(Question
:accessor Question
:type (string 200)
:initarg :Question)
(Answer
:accessor Answer
:type (string 1000)
:initarg :Answer)
(Reasons
:accessor Reasons
:type (string 5000)
:initarg :Reasons))
 (:base-table QUST_ANS))
 (clsql:def-view-class USER ()
((Email_id
:db-kind :key
:db-constraints :not-null
:type (string 50)
:initarg :Email_id)
(Name
:accessor Name
:type (string 50)
:initarg :Name)
(Age
:accessor Age
:type integer
:initarg :Age)
(Gender
:accessor Gender
:type (string 1)
:initarg :Gender)
(Relationship_Status
:accessor Relationship_Status
:type (string 10)
:nulls-ok t
:initarg :Relationship_Status)
(Profession
:accessor Profession
:type (string 50)
:nulls-ok t
:initarg :Profession)
(Interests
 :accessor Interests
 :type (string 100)
 :nulls-ok t
 :initarg :Interests )
(Philosophy
:accessor Philosophy
:type (string500)
:nulls-ok t
:initarg:Philosophy))
(:base-table QUST_ANS))
 (clsql:def-view-class Administrator ()
((Admin_id
:db-kind :key
:db-constraints :not-null
:type integer
:initarg :Admin_id)
(Password
:accessor Password
:type (string 10)
:nulls-ok t
:initarg :Password))
(:base-table Administrator))
```

The DEF-VIEW-CLASS macro is just like the normal CLOS (Common Lisp Object Store) DEFCLASS macro, except that it handles several slot options that DEFCLASS doesn't. These slot options have to do with the mapping of the slot into the database. Only a few of the slot options in the above problem are available.

1) : *column* - The name of the SQL column this slot is stored in. Defaults to the slot name. If the slot name is not a valid SQL identifier, it is escaped, so foo-bar becomes foo_bar.
2) : *db-kind* - The kind of database mapping which is performed for this slot. :base indicates the slot maps to an ordinary column of the database view. :key indicates that this slot corresponds to part of the unique keys for this view, :join indicates a join slot representing a

relation to another view and :virtual indicates that this slot is an ordinary CLOS slot. Defaults to :base.

3) : *db-reader* - If a string, then when reading values from the database, the string will be used for a format string, with the only value being the value from the database. The resulting string will be used as the slot value. If a function then it will take one argument, the value from the database, and return the value that should be put into the slot.
4) : *db-writer* - If a string, then when reading values from the slot for the database, the string will be used for a format string, with the only value being the value of the slot. The resulting string will be used as the column value in the database. If a function then it will take one argument, the value of the slot, and return the value that should be put into the database.
5) : *db-type* - A string which will be used as the type specifier for this slots column definition in the database.
6) *: void-value* - The Lisp value to return if the field is NULL. The default is NIL.
7) : *db-info* - A join specification.

In this database each table as a primary key attribute, which is required to be unique. It is indicated that a slot is part of the primary key (CLSQL supports multi-field primary keys) by specifying the :db-kind key slot option.

The SQL type of a slot when it is mapped into the database is determined by the :type slot option. The argument for the :type option is a Common Lisp data type. The CLSQL framework will determine the appropriate mapping depending on the database system the table is being created in. To determine what SQL type was used for a slot, it could be specified a :db-type option like "NUMBER(38)" and it could be guaranteed that the slot would be stored in the database as a NUMBER(38). This is not recommended because it could make view class importable across database systems. DEF-VIEW-CLASS also supports some class options, like :base-table. The :base-table option specifies what the table name for the view class will be when it is mapped into the database[10].

Inference Engine – It provides the system control. Inference engine is a mechanism that fetches the keywords from the user inputs or queried stored in working memory and matches them with the KB to find out the questions and answers and further continues the session to come to conclusions. This process leads to the solution of the problems. Inference engine also enables the expert systems interface to data sources and to the user.

Spiritual Guru will be a rule-based expert system using forward chaining. In order to execute a rule-based expert system using the method of forward chaining it is needed to fire (or execute) actions whenever they appear on the action list of a rule whose conditions are true. This involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied.

A generic algorithm for this might be:
while values for attributes remain to be input
read value and assign to attribute
evaluate conditions
fire rules whose conditions are satisfied

Several points about this require consideration. First, some conflict resolution strategy needs to be employed in order to decide which rules are fired first. Here method is to fire the rule which the system designer defined first. To cut down on computational time, nothing should be done which does not absolutely need to be done. This means that conditions are only evaluated at the time they might change and that rules are checked (to see if all of their conditions are satisfied) only when they might be ready to be fired, not before. It should be done as attributes are assigned values and should only consider rules and conditions affected by the new attribute assignment. To develop an inference engine for a rule-based system the basic components are:

Attributes: X1, X2 , ... , Xn1
Conditions: C1, C2 , ... , Cn2
Rules: R1, R2 , ... , Rn3
Actions: A1, A2 , ... , An4

It is only needed to execute an action when a rule containing it is fired. A rule is fired only when all of its conditions are satisfied. To detect this, a counter is assigned to each rule and used to keep track of exactly how many of the conditions in the rule are currently satisfied. It is only checked to see if a rule is ready to fire when one of its conditions has become true. In turn, a condition need be evaluated only when all of its attributes have been defined and one has changed. This is to keep track of with a counter assigned to that condition. In addition, an attribute is flagged as defined or undefined [11].

For rule such as:

R1: If X1 = Qid then A1;
R2: If X2 = Reason then A2;
Defined conditions may be:
C1: X1 = Qid;
C2: X2 = Reason;

Then the various lists are set up and the rules and the relationships between the attributes, conditions, rules, and actions may be presented as the graph in Fig. 2.

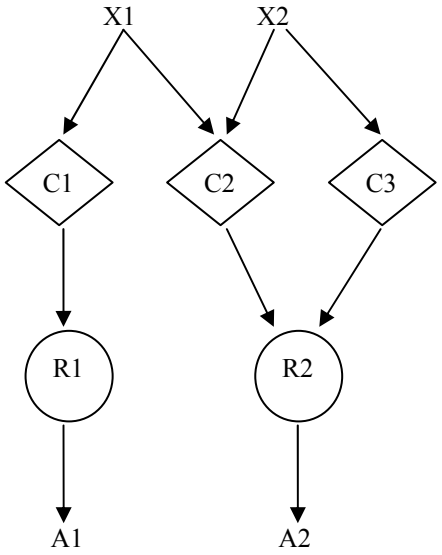

Figure 1. Working of Inference Engine

Inference engine for a system containing the above two rules since the engine operates by doing a depth-first search of the graph, beginning at the attribute being changed and

continuing down the graph whenever the counter assigned to a condition or a rule indicates that all of the information required is present.

Algorithm: Assignment (Xi, v)

PRECONDITION: $X_i \neq v$ and the knowledge base is correct
POSTCONDITION: all appropriate actions have been executed and the knowledge base has been updated correctly
X_i = value v
for each C_j on the condition list of X_i do
if Xi is undefined then increment the counter for C_j
if all attributes needed for C_j are defined then
evaluate Cj
if the value of C_j changed then
for each R_k on the rule list of C_j do
if C_j became true then
increment the counter for R_k
endif
if all conditions in R_k are true then Fire(R_k)
endif
otherwise decrement the counter for R_k
mark X_i as defined
endif
endif
endif

This algorithm is used for assigning a value to an attribute and performing all other appropriate tasks that this assignment triggers [11].

Interface – this ES has graphical user interfaces first one is Input interface, second one is User interface and the third one is Output interface. The Input interface – is admin page. This allows the experts to upload the knowledgebase finals and updating the database. User interface – is user page. This allows for creation of new users and also allows the existing user to consult the expert system in a user friendly manner. Output interface – is result page. Answers of the query entered by the user will be displayed in this page.

B. Tools and Technology

Spiritual Guru expert system will be developed with

Common Lisp - Common Lisp is a programming language particularly suited for AI programs. LISP (LISt Processing) is invented by John McCarthy during late 1950's. CL development & runtime environment behaves like a complete operating system. CL is a well suited for AI programs because of its ability to process symbolic information effectively. The most important feature of CL is its simple syntax and dynamic memory management. It provides features of Object Oriented language. It is both programming language and a powerful CASE tool.

jLinker – This tool is being used because the Spiritual Guru is an web based expert system. The purpose of this tool is to automate the interfacing of Lisp programs to Java class libraries. jLlinker supports a socket interface that allows the Lisp and Java parts of an application to run in separate process, and even in separate hosts. jLinker also supports a native interface that allows the Lisp and Java parts of an application to share the address space of a single process. jLinker allows dynamic, unpremeditated access to the public methods, constructors, and members of Java classes from the Lisp runtime environment. The end result is that the Lisp application may call Java methods as if they were Lisp functions. The documentation of the Java class is all that the Lisp programmer needs to know to use the Java library effectively.

CLSQL - CLSQL is a Common Lisp interface to SQL databases. A number of Common Lisp implementations and SQL databases are supported. The general structure of CLSQL is based on the CommonSQL package by LispWorks Ltd.

VI. Conclusion and Future Work

In current scenario of India the spiritual awareness has grown and people want spiritual guidance and moral confidence for calm and peace full life but don't have time because of their busy life style –this system provides a better solution for them. The personality development programs in the Educational institutes can use this system and it can be a better teacher of life ethics. The limitations of this system are only the limitations of expert systems. 'Spiritual Guru' proposes a reliable solution in place trouble of finding Spiritual experts who are rarely genuine. In future it can be further developed to be accessed by the mobile phones. The mobile platform provides the advantage for person to get consultation practically anytime and anywhere. Animation, sound and video features can be added to behave like Guru because Guru does not provide only facts, but explains and provides solutions in the form of stories and examples. It can also work as a spiritual healer.

References

[1] Fawad Baig, Naima Nawaz and Saif-Ur-Rehman, "Expert Systems For Decision Making In Agriculture Sector", Journal Of Agriculture & Social Sciences 1813– 2235/ 2005 /01 –2– 208 –211 http://www.ijabjass.org.

[2] M. Babita Jain, Amit Jain and M.B. Srinivas, "Web based Expert System Shell for Fault Diagnosis and Control of Power System Equipment", International Conference on Condition Monitoring and Diagnosis, Beijing, China, April 21-24,2008.

[3] Low Tan Jung , Rozana Kasbon, and Hanita Daud, "Mobile Islamic Medication Expert Systems", IADIS International Conference Informatics 2008, Computer and Information Sciences Department, Universiti Teknologi Petronas, Bandar Sri Iskandar, 31750 Tronoh, Malaysia, June 8-11, 2008.

[4] Stephen Hawking, and Bantam Books, "A Brief History Of Time, From The Big Bang to Black holes", Journal of Near-Death Studies,pp. 198, Issue Volume 9, Number 2,Dec 1990.

[5] Sri Aurobindo, "The Life Divine", Sri Aurobindo Ashram, Pondicherry, 1940.

[6] http://www.the-auras-expert.com/

[7] Joseph Giarratano, Gary Riley , "Expert Systems: Principles and Programmings", 2e, TMH New Delhi, 2003.

[8] Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", 3e, PHI New Delhi, 2004.

[9] CpSc810 – Goddard – Notes Chapter 7, http://www.cs.clemson.edu/~goddard/texts/cpsc810/chapA3.pdf.

[10] Kevin M. Rosenberg, Marcus T. Pearce, Pierre R. Mai, "CLSQL Users' Guide" ,onShore Development Inc 2010.

[11] N. L. Griffin and F. D. Lewis , "A Rule-Based Inference Engine which is Optimal and VLSI Implementable", IEEE International Workshop on Tools for Artificial Intellige Architectures, Languages and Algorithms , 23-25 Oct 1989.

Puja Shrivastava is M. Tech. Research Scholar in Computer Science & Engineering Department of Rungta College of Engineering and Technology,Bhilai, Chhattisgarh. Her area of reasearch includes AI, ANN, and information system and security.

Susanta K. Satpathy (M'08) is working as a professor at Rungta College of Engineering and Technology, Bhilai, Chhattisgarh. He received the M. Tech degree in Computer Science and Engineering from NIT, Rourkela in 2003 and B.Tech. degree in Computer Science and Engineering in 1995. He has published above 10 papers in various Journals and conferences. He is life time member of CSI and ISTE since 2008. He worked in various other engineering colleges for about 14 years. His area of research includes Signal processing, image processing and information system and security.

Kapil K. Nagwanshi (M'09) is member of International Association of Computer Science & Information Technology since 2009. He is also member of International Association of Engineers (IAENG) and Computer Science Teachers' Association (CSTA-ACM) Currently he is working as a Reader in RCET Bhilai, and he has published more than 28 research papers in reputed journals, national and international conferences. His research area includes, signal processing, and image processing, and information systems and security.