A knowledge-based system for residential HVAC applications

Two benefits of this expert system for homeowners are reduced electric costs and increased energy savings

By Musa Jafar, Ph.D., A. Terry Bahill, P.E., Ph.D., and Donald E. Osborn

This article describes the development lifecycle, structure, verification, validation, testing and evaluation of the Western Area Power Administration/Salt River Project Residential Expert System (WAPA/SRPRES). This is a residential energy savings advisor, developed at the University of Arizona under a contract for the Western Area Power Administration (WAPA), Boulder City, Nevada, and the Salt River Project (SRP), Phoenix, Arizona.

The expert system was designed to provide readily available knowledge supporting SRP's residential energy audit efforts. The WAPA/SRPRES obtains information about a specific customer's house and energy consumption through the system's interface. This information is obtained directly by having the customer run the system or indirectly by having a utility information specialist run it.

The system then provides specific and easy to follow energy management recommendations that are tailored to the customer's specific needs. Later versions with more depth can be designed for use by trained auditors to enhance their effectiveness and productivity. Other versions equipped with graphics and more detailed explanation and retrieval capabilities can be used as part of a training program for energy auditors.

Background and research motivation

Twenty percent of the power in the United States is consumed by residential buildings, and is distributed as follows: 14 percent for heating and cooling, four percent for water heating and two percent for other functions. This points out the importance of creating energy savings programs for the residential sector that help plug the energy drains of heating and cooling of air and water (Brothers 1988; Brothers, Cooney 1989; Franconi, Dow 1988).

About the authors

Musa J. Jafar is an instructor in computer science at Pima College, Tucson, Arizona. He received a B.S. in mathematics from the Hygazian College, Beirut, Lebanon, an M.S. in mathematics from the American University of Beirut, an M.S. in systems engineering and doctoral degrees in systems engineering and industrial engineering from the University of Arizona.

A. Terry Bahill is a professor of systems and industrial engineering at the University of Arizona in Tucson. He received a B.S. in electrical engineering from the University of Arizona, an M.S. in electrical engineering from San Jose State University, and doctoral degrees in electrical engineering and computer science from the University of California, Berkeley.

Donald E. Osborn is the research engineer for the University of Arizona's Solar & Energy Research Facility in Tucson. He received his B.S. in engineering physics and an M.S. in energy engineering from the University of Arizona. He has over 15 years of experience in solar and energy research, development and engineering.

Homeowners are willing to invest a little time, effort and money to learn more about their home energy usage and how to reduce energy costs. Furthermore, utility companies encourage customers to use their energy more efficiently in an effort to better control utility loads. They are also trying to implement useful residential energy audit programs by distributing booklets, training energy auditors and offering incentives to customers.

In the past, when customers called SRP Customer Services for advice about their energy consumption, they received different recommendations, depending on the energy auditor or information specialist available at the time. Advice that is inconsistent, generalized or hard for the do-it-yourselfer to use is less likely to promote customer action. Such inconsistency motivated WAPA and SRP to seek methodologies to provide a high level of consistent and readily accessible expert advice.

For a residential energy audit program to succeed, much effort, expense and personnel are required to train energy auditors and energy information specialists. There is also a need to provide the customer (directly or through energy auditors and information specialists) with interactive access to in-depth, flexible and tailored expertise without tying up the valuable time of human experts. The advent of powerful personal computers and knowledge-based "expert" systems provides a practical solution to these needs and a more cost effective approach to such residential energy management programs.

Expert systems and knowledge engineering

Expert systems are knowledge-rich, logic-oriented, highly-interactive computer programs. They mimic the input/output behavior of human experts in narrow and specific domains. They use heuristic knowledge to make inferences and explain their line of reasoning. The knowledge might be uncertain or incomplete either because of the randomness of the process or because of lack of evidence to support basic facts during the inferencing process.

An expert system consists of the following four modules:

- An inference engine that is a computer program containing the methods of plausible reasoning to interpret and use the knowledge-base to control the problem-solving scenario. Inference engines use different rule execution methods and inferencing strategies.
- A knowledge-base that is domain specific. It contains the basic facts and inferencing rules that represent the domain heuristics of the system. The knowledge-base is created by the knowledge engineer and captures an expert's knowledge.
- A user interface that contains explanation facilities to support smooth interaction with the user. The user interface plays an important role in establishing the level of performance and acceptability of a system by the end-user. Different production languages (shells) provide different types of user interface. It is the knowledge engineer's responsibility to ensure that the right user interface is provided.

• A knowledge acquisition module that contains strategies to elicit the basic knowledge and aid in building the system's syntax. Some knowledge acquisition modules also check for consistency and completeness of the knowledge-base.

The primary aspects that differentiate expert systems from conventional software programs are separation of the knowledge from the inference engine, user interface facilities, and the knowledge acquisition module.

Experts gain knowledge through sensory contact, education and reasoning. However, an expert's knowledge is often unstructured, complicated and incomplete. Often experts cannot explain their knowledge not only because they cannot verbalize it, but also because they might not be aware of its relevancy to a particular problem.

The performance of an expert system and the resources consumed in its construction are highly affected by the knowledge engineer and his communication capabilities. The knowledge-base is incrementally built through collaboration of the knowledge engineer and the domain expert(s). Communicating with an expert is time-consuming and highly subjective. In addition, important concepts might be overlooked by both the expert and the knowledge engineer in any session. The knowledge engineer has the responsibility of choosing performance measures that can be used to ensure the consistency of the expert, and ensure that the right system is built right.

Another concern of the knowledge engineer is to ensure that all the requirements for building an expert system are satisfied. Some of these requirements are:

• Choosing an appropriate inferencing strategy and knowledge representation scheme (or choosing an appropriate expert system shell);

• Ensuring conformance of the system with the model from which it was built; and

• Ensuring compliance of the system with the requirements that were established for it.

In building this expert system, we found that it was important to designate a primary expert to interact with the knowledge engineer. In the beginning of this project, we worked with many experts individually and in groups. One expert would say one thing and another would say something else (often they were both correct, but were thinking about different specific cases). This was confusing and time-consuming. We finally designated one expert whose responsibility was to talk with all the other experts, derive a consensus, and then talk with the knowledge engineer.

Expert systems applied to energy information efforts

When considering expert systems for utility applications, the adequacy of the problem, the benefits to the utility company and the benefits to the customer should be considered during the concept exploration phase (Brothers 1988; Franconi, Dow 1988). To determine if residential energy management is appropriate for an expert system, we considered these criteria:

• There are identifiable residential energy management experts at the utility who are successful and capable of pointing out appropriate energy saving actions for specific customers to take. The recommended actions are based on factual and heuristic knowledge.

• Energy experts can verbalize their thought processes, and identify what facts, rules-of-thumb and experience they consider when forming their conclusions. It can take time, however, to get to the heart of what they really consider to be important.

It appears reasonable that the area of residential energy management is one where expert systems should perform well. This is because an energy expert can typically provide useful, tailored advice over the phone by questioning the customers about their homes, appliances, energy use patterns and other relevant items, and then analyzing the situation. Furthermore, there is normally a wide range of possible actions that can be taken. Only a few will be most appropriate for the customer to undertake in a particular case.

To determine if the utility could gain from the application of expert systems techniques in this area, several benefits were considered. One key benefit that would be useful to the utility is an increase in employee productivity. An approach that would boost the performance of information specialists and energy auditors would be a tangible benefit.

Energy information specialists and auditors typically have far less training and experience than the energy expert, and their performance level in analyzing an energy problem is typically much lower. However, these specialists and auditors can be provided by the utility in much larger numbers and with far less cost because they require less in-depth training and experience. If with the use of an expert system, the effective performance of the energy information specialists and auditors can be increased to approach that of the typical energy expert, then considerable savings can accrue to the utility.

Finally, the customers' benefits are enormous. In addition to saving them money on utility bills, the system educates them, broadens their knowledge about the sources of energy loss in their houses and helps them establish habits to conserve energy. An expert system would present a homeowner with different options and guidelines to follow and, for some customers, it may identify tax credits, tax deductions or low interest loans for energy conservation.

System development lifecycle

Because of the separation of the knowledge-base from the inference engine, the uncertainty of the requirements and the type of information processed in an expert system environment, the development lifecycle of an expert system is iterative and incremental (Jafar 1989). It starts with an initial prototype that is easy to build. This prototype usually covers a subset of the problem and demonstrates the adequacy of expert systems technology or of providing a solution to the problem. The next prototype is an improved, expanded version of the first prototype.

This cycle of iterative prototyping is repeated until the knowledge engineer and the expert are satisfied with the product, which is not easy to define. At the end of each stage, the knowledge engineer has the responsibility of verifying the syntax, validating the semantics, testing the prototype for conformance to the model and compliance with the current requirements, and working with the experts on evaluating the expert system.

Our first prototype system (developed in three months) was simplistic and naive. Most of the knowledge was extracted from books (Brumbaugh 1983; Edwin 1984; Hedden 1981; Joseph 1971; Reagan 1975; Rosenfeld, Hafemeister 1988). It demonstrated the feasibility of applying expert system technology to solve the problem, educated the knowledge engineer about this specific problem, and facilitated discussions with the experts.

Most criticism of this prototype focused on the difference between the books' methods of giving recommendations and those of the utility experts. Books recommended actions such as shading the cooling system if exposed to direct sunlight. According to the utility experts, such action is hazardous and the payoff is small. Another book recommendation was to move the thermostat if it was exposed to an air current or direct sunlight. The energy experts considered that expensive, with little payoff to the customers who usually do not follow such recommendations anyway.

During the early stages of the project, our designated expert was cautious in criticizing the books. This was understandable because people tend to believe that information presented in books is correct. However, after consulting with other utility experts, she felt more comfortable in disagreeing with the books.

Another shortcoming of book knowledge is its generality. Because they are intended to reach the largest possible audience, books do not meet the needs of specific climate zones, and their recommendations are general. For example, books recommend double-pane windows. However, these windows are expensive, and they are not cost effective in the Southwest where there is little difference between inside and outside temperature most of the year.

The knowledge-base

The conceptual design of the knowledge-base should be based on the user's tasks, requirements and capabilities. The design should minimize the operational and functional obstacles that a user must overcome. The user's tasks can be determined indirectly by talking with the expert or directly through interviews with potential end-users.

Knowledge-based systems rely on knowledge from specific domains to provide solutions to specific problems. Knowledge engineers usually select the best ways of extracting and representing this knowledge, and the software to be used.

Different schemes have been proposed for representing knowledge in expert systems. In production systems (used in WAPA/SRPRES), knowledge is represented as a set of if-then rules (see *Figure 1*). If a rule is triggered and the set of conditions are satisfied, then the set of conclusions will be inferred. Production systems were first introduced by Newell and Simon in the early 1970s for their models of human cognition. If-then rules are used to represent heuristic knowledge. They are easy to understand and seem to effectively reflect the way our utility experts represent and diagnose energy problems.

Another major advantage of this scheme is the visibility of the knowledge, especially because the designer of the expert system will not be the maintainer, which is the case in most software development projects. The designer has to exert extra effort to develop an understandable system to help the maintainers modify and update the system during its useful lifecycle.

During the development cycle of the knowledge-base, several issues were considered. The first was what to include in the knowledge-base. Ideally, we should include every component that contributes to the energy consumption of the house, but is this feasible? Even if it is feasible, what are the advantages and disadvantages of including all items? Will the customer carry out the recommended actions?

Although a customer might be able to say whether the walls have any insulation, a homeowner usually does not know the R-values of the walls' insulation, would not have an easy way to find it (Jackson, Callahan 1981; Joseph 1971), and a specialist would be needed to increase the insulation value. Therefore, a section about wall insulation would probably not trigger customer action.

The second issue was the suitability of the knowledge. Problems such as double-pane glass windows versus single, changing the position of the house thermostat if it is exposed to a heat source, and covering an unshaded cooling system were addressed.

Finally the major energy drains of the house were considered. Heat is transferred between the inside and outside of the house: by conduction through the doors, windows, walls and roof; by convection of air through cracks, holes and joints; and by solar radiation through glass windows, doors and skylights. The amount of energy required for heating and cooling a house can be reduced if the losses are reduced. Air infiltration through loose-fitting windows and doors, cracks in the walls and open dampers can increase the heating and cooling bill by 10 to 20 percent.

Inefficient appliances can also increase energy bills. Regular maintenance and periodic checkup of appliances reduce energy losses, optimize their energy consumption and prolong their lifetimes. For example, the cost of insulating the water tank of a conventional water heater will be recovered in a year or two. Homeowners pay as much as five dollars per month for every 10° the water is heated above 120°F. Draining the water tank of a gas or electric water heater twice a year will cut the energy losses and prolong the life of the tank. Draining the water tank removes the sediment that, if left in the tank, provides an insulation layer between the water and the flame.

if water heater = water heater-W
and type(water heater-W) = gas
and turned down(water heater-W) = no
and turn down message(water heater-W) = M
and displayed(M)
then turn down(water heater-W).

Figure 1. A typical rule. Terms starting with capital letters are used to handle variables in the M.1 production language.

turn off message(water heater-W) =
'While on vacation it is best to turn off an electric water heater. It only takes a couple of hours to reheat the water and you will save the energy that would otherwise be lost when the water heater is not being used for an extended period of time.'

turn down message(water heater-W) =
'While on vacation it is best to turn down the temperature setting of a gas water heater. It only takes a couple of hours to reheat the water and you will save the energy that would otherwise be lost when the water heater is not being used for an extended period of time.'

type(water heater-W) = electric.

Figure 2. A set of facts.

question(turned off(water heater-W)) =
'When you leave the house for more than 3 days, do you turn off the power for your electric water heater?'.

Figure 3. A metafact.

Based on these problems, requirements and issues, and the amount of energy consumed for air heating and cooling, water heating and operating appliances, we divided our system into four categories: air heating and cooling, water heating, air infiltration and shading. These categories represent the major energy saving topics that SRP focuses on in its residential energy saving information efforts.

Each category was subdivided into its different constituent sections. The water heating category, for example, was divided into gas, electric and solar water heating sections. Each section was then further subdivided into its different constituent objects. The gas water heating section, for example, was subdivided into the water tank, thermostat, insulation blanket and pipes. The objects are the major items to which energy-saving actions can be applied. For each object, attributes were defined and values were specified for use in the production rules.

The process of dividing categories into sections and sections into parts maintained the interrelationships between the different objects at the section level, sections at the category level and categories at the system level (whole house).

The WAPA/SRPRES knowledge-base contains 125 rules that captured the experts' knowledge (see *Figure 1*), 80 facts that identified the basic states and outputs of the system (see *Figure 2*), 585 metafacts to control the inferencing and provide a friendly interface with the user (see *Figure 3*), and C-language programs that enabled the customer to save the recommendations into files to be printed. These programs helped reduce the operational hurdles of an end-user. Recommendations were provided based on firing of rules in the knowledge-base in response to facts asserted based on the customer's specific situation.

Insulation of walls or ceiling was not considered in this expert system. Currently, the system focuses on addressing the do-it-yourself type of problems. Remedies that require a specialist's assistance were not addressed.

The reasoning process

The reasoning process used in our expert system is a combination of goal-driven inferencing [backward-chaining; e.g., *goal = check(water heater)*], and data-driven inferencing [forward-chaining; e.g., *whenfound(type(water heater) = electric) = (...)*], as in *Figure 4*. The system starts with a statement of the problem and tries to achieve a set of goals during the consultation.

Figure 4 shows a goal, *check(water heater)*, to be achieved by the system. For each goal, the system first searches the facts in the knowledge-base for values of that goal. If such values are not found, then the system uses rules that have clauses in their conclusions (the *then* parts of the rules) that assign values for the goal. The clauses in the premises of these rules will provide new goals to be achieved.

The process is repeated until no more goals can be generated by applying the production rules and the facts of the system. If the system fails to find rules or facts to provide values for any goals, the user is asked for these values (*Figure 3*). Rules are selected by matching their conclusions to a current goal and fired by evaluating their conditions.

Backward- and forward-chaining were used in conjunction to control the flow of questioning and provide a human-like dialog between the system and the end-user. Forward-chaining was also used to trigger backtracking when the user decided to change answers to previous questions. Backtracking was implemented via an *oops* option that allowed the user to change the answer to a previous question or go back to the start of a given section. (Our term *oops* is derived from the word *whoops*, not from Object Oriented Programming Systems.)

The user interface

User interfaces are important in man-machine systems. The design and implementation of such interfaces are crucial in determining the complexity, final testing and evaluation of an expert system. Our expert system was designed to minimize the cognitive loads on the user by overcoming the operational and functional hurdles.

Following simple instructions allows the user to start the system, end a session, save the set of recommendations, and get a printout. A dialog is carried on with the user through a simple question-and-answer session. For each question, a menu of possible answers is provided. In response to most questions, the user can also select *oops* to change answers to previous questions, *why* to discover why a particular question is being asked or *unknown*.

The *oops* option can return the system to the state it was in before the last response or to the state it was in at the start of the current section of the consultation. As a side-effect (which was not intended during development), the user can repeatedly use the *oops* option to backtrack one level at a time to change the answer to any previous question. The use of the *oops* option is shown in the small section of a consultation with our expert system given in the sidebar.

```
goal = check(water heater)

Whenfound(type(water heater =electric)) =
    [turn off(water heater),
    insulate tank(water heater),
    set temperature (water heater)].

rule1:
if uses separate(water heater)
and type(water heater) = electric
then check(water heater).

rule2:
if whole family(on vacation)
and turned off(water heater) = no
then turn off(water heater).

rule3:
if insulated tank(water heater) = no
then insulate tank(water heater).

rule4:
if dishwasher uses(water heater) = no
and temperature setting(water heater) = M
and M > 125
then set temperature(water heater).
```

Figure 4. The goal statement *check(water heater)* will trigger rule 1, which will cause values to be sought for *uses separate(water heater)* and *type(water heater)*. This is backward chaining. If the value *electric* is found for the expression *type(water heater),* then the *whenfound* statement will sequentially trigger rules 2, 3 and 4. This is forward-chaining. In turn, each of these rules will trigger a backward-chaining inference strategy to find the values of the expressions in their premises (the *if* parts).

Consulting with the expert system

You have already told us that you have both evaporative coolers and air conditioners in your house. Do any of your evaporative coolers and air conditioners cool the same space?

1. Yes
2. No

>> *Yes*

Do alternate between the evaporative cooler and the air conditioner daily, weekly or even monthly?

1. Yes
2. No
3. Oops

>> *Yes*

It is recommended that you use the system in the following manner. Start with the evaporative cooler in the spring when humidity is low. When the monsoons arrive (usually in early July), switch to air conditioning. Then when the humidity drops at the end of the summer (usually mid-September), go back to evaporative cooling. We strongly discourage alternation between the two systems even on a weekly basis. Moisture build-up in the rooms will add an extra load when you use the air conditioner and cancel any savings you might have gotten from using the evaporative system.

You have already told us that an evaporative cooler and a central air conditioner share the same ducts. Do you close the damper of the evaporative cooler when you shift from evaporative cooling to air conditioning?

1. Yes
2. No
3. Oops

>> *Yes*

Does the whole family frequently leave the house for more than five continuous hours?

1. Yes
2. No
3. Oops

>> *Oops*

Which of your previous answers do you want to change?

1. Last answer only
2. All answers for the current section
3. None

>> *1*

You have already told us that an evaporative cooler and a central air conditioner share the same ducts. Do you close the damper of the evaporative cooler when you shift from evaporative cooling to air conditioning?

1. Yes
2. No
3. Oops

>> *No*

Close the damper of the evaporative cooler whenever you operate the central air conditioner. Otherwise cooled air will escape through the evaporative cooler.

The implementation of the *oops* option was not trivial. The procedures had to keep track of the changes that each user's response brought to the intermediate conclusions. They also had to undo the effects of these responses (on user's request), reset the affected components of the system, and keep smoothness of transition between states during a consultation.

The *oops* procedures required extensive debugging and developmental testing. Earlier prototypes of our system implemented an *oops* option that presented a menu of all previous questions for possible correction. Such an implementation was hard to maintain and the size of the knowledge-base grew exponentially with each iteration during the development process.

The production language used (the M.1 expert system shell) does not support saving fragments of the consultation and appending them into one file without overwriting existing data in the file. Therefore, we wrote C-programs to interface with the production language, save the sets of recommendations to a file and assist the user in printing them.

System testing, verification and validation

Testing, verification and validation are the most important aspects of expert system design and there are now tools available to help with these procedures (Bahill, 1990). Testing merely shows errors (if any) in a system; it does not explain the cause of errors.

Testing usually detects three types of errors: logical errors related to the knowledge-base; syntactic errors; and missing knowledge. We used panels of evaluators from WAPA and SRP for developmental and final testing. We used feedback from these evaluators and worked with the designated expert to improve the system.

The system was also tested during a three-day computer show in a shopping mall. The users were mall visitors attending the show; they used the system without any help from the designers. They were generally happy with the system's performance, and felt the information provided was useful.

Verification and validation of the system were performed during and at the end of each stage of development. This was done with a program developed as part of a Ph.D. dissertation (Jafar 1989). The program checked the knowledge-base of the expert system for internal inconsistencies, multiple methods of obtaining values, dead-end rules, referential integrity constraints, and mismatches in the knowledge-base. The program also created knowledge-base dictionaries of the objects, their attributes and the corresponding values. The program verified and validated the knowledge-base interactively. It presented the developers with a set of possible violations, and it was the developers' responsibility to maintain the validity of the knowledge-base and carry out the changes.

Conclusions

One goal of our project was to give WAPA and SRP personnel experience with expert systems so they could explore how such techniques might be employed by utilities to enhance productivity and ameliorate problems. We succeeded in providing an introduction to expert systems and knowledge engineering through an exercise that involved WAPA and SRP personnel and resulted in a useful example. The expert system has to undergo further testing, allowing the utilities' employees to run the system and to provide feedback and evaluation.

The system will eventually be used by the utility companies to serve their customers. The utilities' telephone specialists will use the system to answer customers' questions, and customers can use the system during their visits to the utilities' electric shops in their service area. ■

References

Bahill, A.T. 1990. *Verifying and Validating Personal Computer-Based Expert Systems.* Englewood Cliffs, New Jersey: Prentice-Hall Inc.

Brothers, P., Cooney, K. 1989. "A knowledge-based system for comfort diagnostics." *ASHRAE Journal*, September.

Brothers, P. 1988. "Knowledge engineering for HVAC expert systems." ASHRAE *Transactions*, Vol. 94.

Brumbaugh, E. 1983. *Heating, Ventilating and Air Conditioning Library.* New York: Bobbs-Merrill Corporation.

Edwin, A. 1984. *Air Conditioning: Home and Commercial.* New York: Bobbs-Merrill Corporation.

Energy Saving Projects for the Home. 1980. San Francisco, California: ORTHO Books.

Fire Places and Wood Stoves. 1981. Alexandria, Virginia: Time-Life Books.

Franconi, E., Dow, J. 1988. "PC-Expert system applications in residential auditing." *Proceedings of the 3rd National Conference on Microcomputer Applications in Energy.* Tucson, Arizona.

Hedden, J. 1981. *Heating Cooling Ventilation Solar and Conventional Answers and Projects for the Home Owner.* Passaic, New Jersey: Federal Marketing Corporation.

Jackson, D., Callahan, J. 1981. "Energy conservation with microcomputers." *Byte*, July.

Jafar, M. 1989. "A tool for interactive verification and validation of rule based expert systems." *Ph.D. dissertation.* Tucson, Arizona: University of Arizona.

Joseph, O. 1971. *How to Design Heating-Cooling Comfort Systems.* Birmingham, Michigan: Business News Publication Company.

M.1 Reference Manual. 1986. Palo Alto, California: Cimflex Teknowledge.

Reagan, J. 1975. *Energy Conservation For the Home or How to Lower Your Utility Bill.* Tucson, Arizona: Electrical Engineering Department, University of Arizona.

Rosenfeld, A.H., Hafemeister, D. 1988. "Energy-efficient buildings." *Scientific American*, April.

Solar Remodeling Passive Heating and Cooling. 1981. Menlo Park, California: Sunset Books, Lane Publishing Corporation.

Thumann, A. 1988. *Optimizing HVAC Systems.* Lilburn, Georgia: The Fairmont Press Inc.