Expert System Implemented For Problem Solving In Commerce/Business Domain

Viral Nagori, Dr. Bhushan Trivedi

ABSTRACT:

The application of Expert Systems in commerce/business domain is relatively new research area. The rapid growth of artificial intelligence has lead to the development and implementation of expert systems for the purpose of commerce/business problem solving. This generated a need for analysis and review of Expert Systems in commerce/business domain. Here a few expert systems already implemented in the area of resource allocation/ space utilization are reviewed. In addition, analysis is provided based on the following factors: Objective of system, need for the system, development environment, heuristics used, user interaction, knowledge base, explanation facility, inference engine, knowledge update facility, performance evaluation, and category of expert system. At the end, comparisons between expert systems based on the above factors are provided. Though complete comparison is not possible, because each expert system is supposed to be expert in its own domain and work in the specified domain with given constraints. Still comparison enabled me to identify the factors which should be considered for the development of expert system in human resource area. The main objective of the analysis and review is to be acquainted with knowledge to develop Expert System in other domains. At the end of the paper, guidelines for developing Expert Systems are presented.

Keywords: Expert systems, Explanation facility, Inference engine, Knowledge update facility, Resources allocation.

1. INTRODUCTION:

Expert systems are used to perform a variety of extremely complicated tasks that could be performed only by a limited number of trained human experts. Expert systems are used in a wide variety of applications like diagnosis, planning, forecasting, design, control etc (David Rolston, 1988). Most of the expert systems so far developed are in the area of science and technology. The expert systems are implemented in the various disciplines like maths, chemistry, biology, medicine, and physics.

As part of research study, I wanted to develop an Expert System in the HR domain. However, so far I couldn't come across an Expert System in that domain. Therefore, I continued my research for Expert systems developed or implemented in broader related domains. The main objective here is to understand the Expert Systems and acquire the knowledge to develop an Expert System in other domains. Very few expert systems are developed and implemented in the areas of commerce/business domain like finance, customer service, resources allocation etc.

The paper here reviews three expert systems already implemented for problem solving in business/ commerce domain in the area of space utilization/ resource allocation. In addition, analysis is provided based on the following factors: Objective of system, need for the system, development environment, heuristics used, user interaction, knowledge base, explanation facility, inference engine, knowledge update facility, performance evaluation, and category of expert system. The factors are explained below.

Objective of the system: What an expert system is trying to achieve when implemented.

Need for the system: Why expert system development is needed? What problems it is trying to solve?

Development environment: The platform, tools and technology used to develop and implement expert system.

Heuristics used: The general rule of thumb, which can be programmed.

User Interaction: Does expert system provide any window for user input or its just automated system, which runs in batch mode.

Knowledge base: It means the storehouse for the knowledge like the rules, facts, procedures and heuristics. It acts as a database for expert system.

Explanation facility: An expert system must have ability up to certain extent the reasons, which lead to decision.

Inference engine: Can expert system infer new knowledge from the existing knowledge base?

Knowledge update facility: The rules for the domain keep changing rapidly in today's environment. The expert system should be able to generate new knowledge automatically from the experience. In short it should be able to self learn and self update.

Performance evaluation: Does the expert system meet the objective for which it is developed? Can it meet the performance of the expert in the field?

Category of Expert System: The expert systems are classified into three categories: Assistant, colleague, and Expert. (Davis, 1985)

Three expert systems are reviewed based on the above factors. The first one is ALEX, which is an Expert System for truck loading. Second, one is resource allocation for flight crew assignment. And the third one is time tabling of maths course in university for graduate course.

The reason behind choosing ALEX is to understand the optimum utilization of space area for loading packed boxes in the truck. Resources allocation and space utilization are the most common problems faced by any business house. And coming with the feasible solution requires use of human expertise. The Expert System in the area timetabling of maths course can be implemented for any course in any university.

2. ALEX – AN EXPERT SYSTEM FOR TRUCK LOADING: (RON LEMASTER, 1990)

Definition: It plans the loading of boxed products in the production data processing environment of a large manufacturer of household appliances.

Objective: The system seeks to optimize the volumetric utilization of a trailer while minimizing the likelihood of damage to the products while in transit.

Need for the system:

1) The cost of transporting products from their place of manufacturer to distribution centres and sales outlets often represents a significant portion of the product's total cost.
2) Damage to products while in transit can be costly.
3) Manufacturers are interested in utilizing as much of the space in the transport vehicles as possible while arranging the items in each load.

Development Environment: ALEX was prototyped in ECLPS (Enhanced Common Lisp Production System). Moreover, the production implementation was done in Knowledge Tool and PL/I to run on an IBM mainframe. The operating environment is CICS, a transaction processing subsystem running on the MVS operating system. It is developed by IBM. ALEX has 18000 lines of knowledge tool and PL/I code.

Heuristics used:

1) Group boxes of similar product type, size and weight into larger, composite objects.
2) Pack the load one row at a time. Thus, it reduces problem of loading trucks into simple two-dimensional problems.
3) Always pack as tightly as possible towards the front of the trailer and against the right hand wall.
4) Determine which product to go in certain zones of the trailer. Certain products are more prone to damage than others are.

5) Identify the products, which will make a good wall at the end of the load to hold the boxes in front of them in place. E.g. Refrigerators.

User Interaction: Clerks communicate with Alex through conventional interactive screen on display terminals. Every night it runs through batch job stream.

Knowledge Base: ALEX uses formal rules and policies of the manufacturer as well as heuristics and strategies gathered from the company's best loaders to develop volumetrically efficient load plans that also minimize the likelihood of in transit damage. Thirty-five test cases were observed where truck was loaded by experts.

The examples of the knowledge base are: certain products can be stacked only so high, certain products can be placed in specific zones of the trailer to get the smoothest ride, padding materials must be placed between certain products to prevent them damaging each other, refrigerators are placed at the rear of the load to prevent other boxes from shifting. This can be expressed as rules.

Each expert loader had its own personal style and unique approach to certain problems of truck loading. However, the basic framework was uniform.

Explanation Facility: ALEX does not have any explanation facility. It only generates a printable load plan for use by forklift drivers at the loading docks.

Inference Engine: Based on the plans and orientation it receives for each boxes for lading the truck, ALEX had been able to infer new knowledge.

Knowledge Update Facility: ALEX was not able to automatically generate new knowledge and update its existing knowledge base.

Performance Evaluation: Alex was continuously tested against 35 test cases. The ALEX was able to equal or better the volumetric efficiency of the loaders in more than half cases. The complete planning of an average load takes less than three CPU seconds on and IBM 3090.

The reasons for the failure in half of the cases were the constraints on which ALEX was developed. Experts bend the rules in many situation, Which ALEX cannot do. For example, the constraint was to leave at least 2 inches of space between each row and the left hand of the trailer, and one inch between the top of any row and the ceiling.

Category of Expert System: ALEX can be considered as colleague, as it is performing significant subset of an expert system.

Summary: ALEX was intended to raise the level of performance of the majority of the company's loaders and to bring more uniformity to the loading procedures throughout the enterprise. If Knowledge update facility and explanation facility can be provided then it can be become truly Expert System on a larger scale, which can be used for loading trucks in any enterprise.

3. EXPERT SYSTEM FOR RESOURCE ALLOCATION : FLIGHT CREWS ASSIGNMENT: (EHUD GUDES, TSVI KUFLIK, AMNON MEISELS, 1990)

Definition: It assigns crews to air borne missions and creates a weekly assignment of crews to their activities.

Objective: The system tries to provide feasible solution that satisfies the problem's real-life constraints, rather than looking for an optimal solution by using simplified constraints.

Need for the system: Resource allocation problems are very hard to solve exactly or optimally. These problems can be solved by human experts who have gained much experience and intuition into their own specific problem domain.

For solving these problems, experts use heuristics and rules of thumb, which are very difficult in mathematics to formulize.

It is very difficult to formalize all complex constraints. Relaxation of constraints is not possible within the framework of the expert system approach.

Sometimes situations demand last minute changes to the general allocation plan. E.g. In case of sick pilot that is scheduled on a specific morning for a flight.

Earlier programmers have created a tool like DSS to help a human in allocating resources, rather than trying for an expert system that solves problem completely. However, such tools have lot similarities to Expert Systems.

Development Environment: The Expert System shell is developed using Prolog. Dependency directed backtracking is used in the approach by the developer. Control strategy as well as backtracking strategy is hardcoded in the Prolog program.

Heuristics used:

Both activities and resources are assigned the priorities.

Each resource must be allocated or must perform certain number of activities per week.

Certain restricting rules must be followed. Like, crewmember has to have certain relaxation time-slice between two assignments.

Assign higher priorities to activities which are scheduled to problematic time of day (nights or weekends.)

Try to allocate crewmember with low availability first.

Every crew must contain crew members with the right or higher qualifications for the roles they are assigned for.

Crewmembers must not be assigned to intersecting activities.

User Interaction: A user interface allows running the resource allocation in batch or interactive mode. In an interactive mode, it is possible to control both forward allocation and backtracking policy. An expert system allows relaxing constraints when backtracking fails.

A user interface enables output in a tabular or graphical format which presents resource allocation in a pleasant and understandable manner.

Knowledge Base: A knowledge base editor which enables the user to enter names of resources, names of activities, priorities, time periods, recommending rules, and rules describing constraints. The editor also allows defining backtracking rules.

Explanation Facility: The explicit explanation facility was missing from the system. However, system allows to stop backtracking and start the constraint relaxation of lower priority constraints.

Inference Engine: An inference engine allocates resources in a forward manner, and can follow recommending rules during allocation. The inference engine checks the satisfaction of local constraints. The back tracking facility of this engine follows the rules of local change policy.

Knowledge Update Facility: An Expert system here was not able to automatically generate new knowledge and update its existing knowledge base. The knowledge base editor is used to enter new knowledge and constraints.

Performance Evaluation: The Expert system was tested with 16 activities, 27 crew- members, and altogether 48 slots. The performance of the system was tested in four different versions. The first version performed allocation with full constraints. Second version performed allocation without the use of priorities for activities and resources. The third version performed allocation without using any local change policy. The fourth version did not assign priorities and did not

use local change policy, which is the worst-case scenario of trying a resource allocation program without using any expert knowledge at all.

The results prove the importance of backtracking strategy. The control part that performs local changes brings the size of the search space down by a factor of at least a thousand.

Category of Expert System: The expert system can be considered as colleague, as it is performing significant subset of an expert system. However, it is not complete expert system as explanation facility and knowledge update facility was missing.

Summary: The implementation of expert system was proved very successful and efficient. The system here comes out with feasible solution which can be implemented rather than just optimize solution without any constraints. The resource allocations were made in few minutes of runtime by Prolog based systems.

4. EXPERT SYSTEM FOR RESOURCE ALLOCATION : TIME TABLING OF MATHS COURSE: (EHUD GUDES, TSVI KUFLIK, AMNON MEISELS, 1990)

Definition: It creates a weekly timetable for different populations of students taking maths courses. There are nonempty intersections between the different populations.

Objective: The system tries to provide feasible solution that satisfies the problem's real-life constraints, rather than looking for an optimal solution by using simplified constraints.

Need for the system: Resource allocation problems are very hard to solve exactly or optimally. These problems can be solved by human experts who have gained much experience and intuition into their own specific problem domain.

For solving these problems, experts use heuristics and rules of thumb, which are very difficult in mathematics to formulize.

It is very difficult to formalize all complex constraints. Relaxation of constraints is not possible within the framework of the expert system approach.

Sometimes situations demand last minute changes to the general allocation plan. E.g. In case of faculty who is on leave for a week.

Earlier programmers have created a tool like DSS to help a human in allocating resources, rather than trying for an expert system that solves problem completely. But such tools have lot similarities to Expert Systems.

Development Environment: The Expert System shell is developed using Prolog. Dependency directed backtracking is used in the approach by the developer. Control strategy as well as backtracking strategy is hardcoded in the Prolog program.

Heuristics used:

- Both activities and resources are assigned the priorities.
- Each resource must be allocated or must perform certain number of activities per week.
- Certain restricting rules must be followed. Like, Faculty must not have more number of lectures than desirable by rules.
- Assign the same time slots to courses that are taken by different population of students.
- Assign time slots according to teacher's constraints.
- Do not assign the same time slots to courses given by the same teacher.
- Do not assign the same time slots to graduates courses and labs.
- In case of clash, look for an already assigned course with high flexibility and small number of student populations, replace its time slots.

User Interaction: A user interface allows running the resource allocation in batch or interactive mode. In an interactive mode, it is possible to control both forward allocation and backtracking policy. An expert system allows relaxing constraints when backtracking fails.

A user interface enables output in a tabular or graphical format which present resource allocation in a pleasant and understandable manner.

Knowledge Base: A knowledge base editor which enables the user to enter names of resources, names of activities, priorities, time periods, recommending rules, and rules describing constraints. The editor also allows defining backtracking rules.

Explanation Facility: The explicit explanation facility was missing from the system. However, system allows to stop backtracking and start the constraint relaxation of lower priority constraints.

Inference Engine: An inference engine allocates resources in a forward manner, and can follow recommending rules during allocation. The inference engine checks the satisfaction of local constraints. The back tracking facility of this engine follows the rules of local change policy.

But certain constraints are unbreakable, which makes inferring new knowledge very difficult by an Expert System.

Knowledge Update Facility: An Expert system here was not able to automatically generate new knowledge and update its existing knowledge base. The knowledge base editor is used to enter new knowledge and constraints.

Performance Evaluation: The Expert system was tested for efficiency. The results were evident and essential for the successful solution of the course timetabling problem.

Category of Expert System: The expert system can be considered as colleague, as it is performing significant subset of an expert system. However, it is not complete expert system as explanation facility and knowledge update facility was missing.

Summary: The implementation of expert system was proved very successful and efficient. The system here comes out with feasible solution which can be implemented rather than just optimize solution without any constraints. The resource allocations were made in few minutes of runtime by Prolog based systems.

5. COMPARISION BETWEEN EXPERT SYSTEM BASED ON ABOVE FACTORS:

Factors	ALE	Flight Crew Assignment	Time tabling of maths course
Development environment	ECLPS Language	Expert system shell and Prolog language	Expert system shell and Prolog language
Heuristics	Defined	Defined	Defined
User Interface	Limited user interaction	Interactive mode	Interactive mode
Knowledge base	based on heuristics	allow user to create using editor	allow user to create using editor
Explanation facility	Absent	Absent	Absent
Inference engine	Able to infer new	Provide back	provide back

	Knowledge	tracking	tracking a and forward allocation
Knowledge update facility	Absent	Absent	Absent
Performance	50% Success	Pass through all testing	Efficiency was achieved

6. CONCLUSION:

The review and analysis of the above-mentioned Expert Systems suggests that we can achieve a feasible solution of the real life problems in the area of Business/Commerce Domain by use of Expert Systems, which requires human expertise to solve the problem. All the Expert systems reviewed here have been able to bring solution along with the enforced constraints. All the Expert systems discussed here are able to run on interactive mode and batch mode both. The business rules can be coded in knowledge base. However, the above Expert Systems discussed here do not have explanation facility and knowledge update facility, which is handled manually. If those can be added, then the system can be considered as a real Expert System rather than Colleague or Assistant to Experts.

From the review and analysis mentioned above, I have been able to frame the guidelines to develop Expert System in Human resource domain to solve the real world problems like employee retention in IT Industry. The guidelines are as follows:

1) The Expert system should try to arrive at feasible solution with all the constraints rather than optimum solution with no constraints.

2) If forward chaining fails to bring the solution than the system should have the facility of back tracking to reach the solution.

3) The hard constraint can be coded as rules or heuristics in knowledgebase.

4) The constraints should be given priority: High or low. If back tracking fails then the Expert System should be able to relax the low priority constraints.

5) The idle expert system should run in both modes: interactive as well as batch mode.

6) Inference engine should be able to update existing knowledge base with the new knowledge.

7) The Expert System should have explanation facility for the solution it generates.

8) A true Expert system should be able to attain feasible solution with all the constraints in the worst-case scenario.

7. REFERENCES:

1. David Rolston, (1988), "principles of artificial intelligence and expert systems development", Tata McGraw Hill, Chap-1, page 1.
2. Davis, (1985), "problem solution with expert system", Texas instruments, Dallas.
3. Ehud Gudes, Tsvi Kuflik and Amnon Meisels, (1990), "An Expert System Based Methodology for Solving Resource Allocation Problems", ACM 089791-372-8/90/007/0309.
4. Ron Lemaster, (1990), "ALEX- An Expert System for Truck Loading", IBM Commercial Systems Integration Boulder, ACM 089791-372-8/90/0007/0638.

AUTHOR'S PROFILE

Mr. Viral Nagori is working as an Asst. Professor in GLS Institute of Computer Technology from the last 7 years and having total experience of 8 years in academic. He is currently purusing PhD in design of Expert system prototype for analysing motivational strategies on ICT Human Resources. He has published 7 papers in national and international journals. Other areas of interest include time management, stress management, body language, and communication skills.

Dr. Bhushan H. Trivedi, Ph.D. is currently working as a Director in GLS Institute of Computer Technology. He has more than 20 years of academic experience in graduate courses primarily MCA (Master of Computer Applications). Apart from pedagogy and effective teaching methods, his other research interests are Information Security and application of mobile agents in distributed intrusion detection as well as prevention systems and wireless sensor networks.