EXPERT SYSTEMS INTERPRETING COLLECTIVE AGREEMENTS IN THE BUILDING INDUSTRY

Jolkkonen, Esa & Salokivi, Jouni

Technical Research Centre of Finland (VTT)
Laboratory of Urban Planning and Building Design
Section of Construction Management and Economics
Itätuulentie 2
SF-02100 Espoo
Finland

ABSTRACT

This paper presents an expert system project, the main aim of which was to obtain information on the viability of the present micro computer-based expert system shells in interpreting a collective agreement of the building industry in Finland. The same expert system was built using two different shells, Guru and Insight 2+, in order to compare them and to study their specific features. An important aspect of the project was the methodology of transferring an expert system developed with one shell to another shell.

The domain of sickness pay as a part of the collective agreement was chosen to be an appropriate area for an expert system since the interpretation of the agreement is complicated by many exceptions and special cases. It is also a domain which constantly raises many problems in the building industry as disputes over proper pay arise both on-site and in the main office.

1. Introduction

In the Technical Research Centre of Finland, Laboratory of Urban Planning and Building Design, an expert system project for interpreting the collective agreement of the building industry was started in August 1986. This project is part of a larger endeavour called Expert systems on the building site. In this first phase two expert systems have been developed with different expert system shells for determining the sickness pay of the employee. The project has been carried out in co-operation with The Federation of the Finnish Building Industry which has both acted as the client and provided the expertise for the systems.

2. Legal expert systems in general

Law as an area for expert system development is drawing much interest. The legal area however, poses some special problems for expert system development. Modelling what a lawyer does is more difficult than modelling what experts do in many technical or scientific domains due to the complexity of the legal process, the need for commonsense reasoning and the lack of a thorough model of the legal process.

Legal expert systems can be grouped into four important types of applications: case management, monitoring, legal interpretation and document generation (Table 1.) /5/.

Application areas for law	*Systems description*
Case management	Organizes case information, estimates case value, and strategies for negotiation and case disposition.
Scheduling monitoring and retrieval	Schedules attorneys' activities, monitors legal data bases and knowledge bases to find changes in the law that could affect active cases, and assists in retrieving knowledge from these sources.
Interpretation and prediction	Interprets the law (e.g., statutes, regulations) in the context of a particular question or problem, anticipates the legal consequences of proposed actions, and predicts the effects of changes in legislation.
Document generation	Produces legal documents (e.g., wills contracts, draft legislation) by selecting or composing appropriate pieces of text and organizing them into document form.

Table 1. Important application areas for expert systems in law.

The expert systems built in this project are an example of legal interpretation application.

3. Objectives of the expert system project

The project started with two major objectives. One was to study the viability of the present micro computer based expert system shells in interpreting a collective agreement and the best ways along which to proceed in this domain in the future. The second objective was to develop a functioning expert system that can be deployed for everyday use in the building industry. In order to compare expert system shells and to study their specific features the same expert system was built using two different shells, Guru and Insight 2+. An important aspect in connection with this was the methodology of transferring an expert system developed with one shell to another shell. One of the ulterior purposes of the project was also to gain practical experience in different expert system development methods.

4. Choosing the domain for the expert system

The collective agreement of the building industry was analyzed in close co-operation with the experts from The Federation of the Finnish Building Industry in order to find the most suitable application for the expert system project. The domain should be both representative and complex enough in order that in developing the systems enough relevant experience of the whole collective agreement field would be gained in respect of the needs for future knowledge based representation. Since the final system was intended for productive use, the chosen domain also needed to be of everyday interest.

Of the many different issues in the collective agreement that were considered, the sickness pay of the employee was chosen to be the most appropriate for this project. The experts were of the opinion that if the expert system techniques were able to solve this, the most difficult issue, then other related problems should pose no serious obstacles.

The following criteria were especially in favour of choosing the subject of sickness pay:

- The question raises many problems in the building industry as disputes between the employees and employers, over proper pay, arise both on-site and in the main office
- The rules concerning the determination of the pay are complicated and in special cases difficult to comprehend
- The rules in the collective agreement and their interpretation change yearly, which makes knowledge management difficult. The knowledge has to be sought from many different sources, e.g. collective agreements, laws, court cases, circulars, databases and experts

- There are many special aspects within a single sickness case which affect both each other and the final solution. Thus, the number of possible cases is almost infinite.

A micro computer based approach was chosen in favour of an expert system in a mainframe computer because:

- Micro computer based shells are much cheaper than large expert system shells and can therefore also be used in small building companies
- They are better suited to rapid prototyping and experimentation
- They can be more easily integrated with the existing information processing technology and system.

5. Present situation in determining the sickness pay

For the moment, clerks at main offices and site managers on-site are responsible for calculating the sickness pay of the employees. An experienced clerk can solve normal cases rather readily but it takes much longer to solve the more difficult cases. In many such cases, the arguments for the decisions are not clear and therefore neither are the solutions as accurate as they might be. Consequently, in order to remain on the 'safe' side, employers often pay out more sickness pay than they are obliged to by the conditions of the collective agreement. In complicated cases, either experts from The Federation of the Finnish Building Industry must be consulted or the decisions must be based on a simplified solution, which similarly results in excess pay. Some cases occur so seldom that even if they have once been solved, the solution no longer remains in the mind of the responsible party.

6. General description of the sickness pay expert system

The expert system takes into account all the facts concerning sickness pay and it is able to give information on the amount of pay, on those working days for which the payment is to be made, as well as any other aspects relevant to the assessment (e.g. the effects of holiday leave).

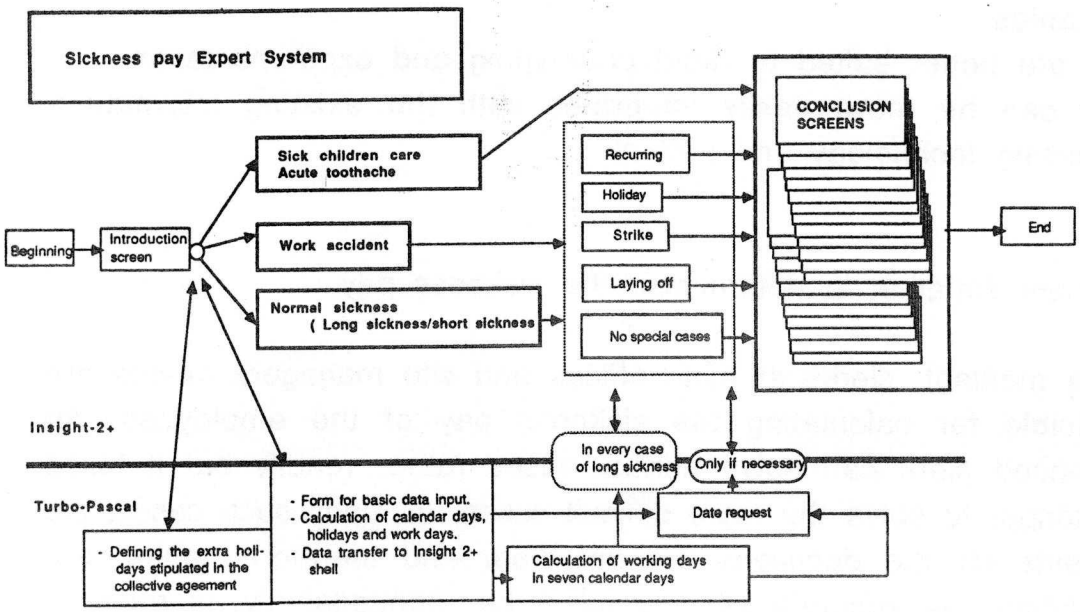

Figure 1. General scheme of the expert system.

The rules in the expert system are based on the valid collective agreement, current legislation (e.g. social security laws), court decisions and The Federation of the Finnish Building Industry's databases on agreed interpretation.

6.1 The Guru and Insight 2+ expert system shells

Guru is a system which has in addition to the expert system shell other tools for system development (e.g. database management, spreadsheet calculation). It offers good possibilities for tailoring the user interface to suit the problem and has a direct connection between the expert system part and the conventional algorithmic calculation in the Guru programming language. In data input, forms and formulae can be used efficiently and there are built-in programs in Guru for converting dates into integer numbers and vice versa. The system is very open, thus enabling all parts of it to be interactively used in direct connection with each other.

Insight 2+ is a compact expert shell with a straightforward language for the knowledge base: everything is represented by rules and therefore the environment in Insight 2+ is more closed than in Guru. The shell offers good possibilities for joining external programs to the expert system, which makes it possible to use the more efficient programming languages for time consuming calculations. Since user interface cannot be changed, the user either has to be content with the shell's standard interface or he needs to custom build one by means of a conventional programming language.

6.2 The Guru and Insight 2+ expert system versions

The first version of the sickness pay expert system was built using the Guru expert system shell. Not all of the knowledge in the system could be presented in the expert system's rules, of which there are about 50. Since a good deal of date conversion and calculation is needed, a significant part of the knowledge had to be embedded in the calculation routines in the Guru programming language. Tailoring the user interface also had the effect that part of the knowledge needed to be placed outside the rules.

The main principle behind building the application using Insight 2+ was that as much as possible may be represented by rules and that traditional programming is only required where rule technique is not feasible. With all its features, the expert system has about 300 rules and the whole system can be printed out as a knowledge tree - which helps in the documentation and correction of errors. All the calculations and part of the user interface (input forms) are programmed using Turbo-Pascal. A user friendly calendar featured at the end of the program was also made by means of Turbo Pascal (Figure 2.). The calendar format represents the way in which clerks and site managers are currently used to counting the sickness pay.

SEPTEMBER 1986

8 M	9 T	10 K	11 T	12 P	13 L	14 S	15 M	16 T	17 K	18 T	19 P	20 L	21 S	22 M	23 T	24 K	25 T	26 P	27 L	28 S	29 M
S	1	2	3	4	5	-			S	1	2	3	-	4	5	6	7				
X	X	X	X	X					N	X	X	-	-	X	X	X	X	Y	-	-	Y

The sickness pay is paid:

N = normal wage
X = at the most 38.00 FIM per hour
Y = at the most 50.00 FIM per hour

Figure 2. An example of the user friendly calendar for defining the sickness pay.

The present status of the systems is such that the Guru version is completed and has been in test-use in The Federation of the Finnish

Building Industry. A continuation project has been initiated with the aim to develop the Insight 2+ version further so that it may be more comprehensive and carry more features than the Guru version. This project will reach completion by the beginning of the summer 1987.

6.3 Evaluation of the shells and versions

Despite both expert system shells being rule based, there were distinct differences in the development work between them (see Figure 3.). Insight 2+ seems to be better suited to team work due to the clearer structure of its knowledge base. Since Guru offers greater possibilities and temptations for writing rather more complicated knowledge bases and programs, it is more suited to the individual than to a group.

The Insight 2+ version is quicker than the Guru version because the calculations are carried out by a more efficient programming language rather then the slower Guru programming language, which cannot be compiled. It is also easier to learn to use the Insight 2+ shell than the Guru shell and as a result of this Guru cannot be recommended as a beginner's tool for building expert systems. On the other hand, Guru has more possibilities and tools for sophisticated system development.

Insight 2+

```
RULE      Holiday's effect on the sickness pay
IF        The sickness has begun before the holiday
AND       The sickness has lasted longer than 7 days
AND       The employee wants to move his holiday
THEN      Normal sickness pay is paid and the employee's
          holiday must be moved
```

Guru

```
RULE:     R.1.13.1
IF:       holtail = "holiday" and sickhelp = true and
          b1 - a1 + 1 > 7 and holnumb >= b1 and wantmo = true
THEN:     perform "ask7days";
          obs3 = "The first 7 holiday days are self risk days.";
          succeed = true;
COMMENT:  When the sickness time has been more than 7 days
          during the holiday time, the sickness pay is paid
          only for the time over these 7 days. ASK7DAYS
          subtracts these 7 days from the total number of
          days sick. The employee wants to move
          his holiday because of the sickness.
```

Figure 3. Example of the same rule in Guru and Insight 2+.

7. Experience from the development work

The methodology for building the expert systems in this project was much the same as described in the expert systems literature (see

Figure 4.). One exception being that the knowledge acquisition process took much longer than expected and continued in fact almost throughout the whole project. Partly for this reason therefore, the testing (referred to in Figure 4. under 'development') also proved a longer process than anticipated. It seems that the various stages in building an expert system are even more overlapping than Figure 4. indicates.

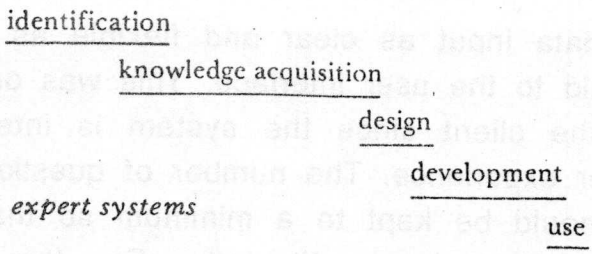

Figure 4. The stages in building expert systems /1/.

One of the most difficult phases during the project was the implementation of the calculation of dates and time intervals. The Finnish calendar contains some special holidays which make the calculation even more complicated. In the collective agreement there are special holidays for reducing the annual work time, which can be agreed on either at company level or even concerning single employees. Another intricate part was to determine and calculate the different cases concerning the recurrence of a previous sickness. This was a task which though relatively easy (if a little time consuming) to do manually on a calendar, was found to be rather complex and difficult to implement using the expert system shell and common computer languages. The problem was solved in the Guru version by using the Guru programming language and in the Insight 2+ version by using the Turbo-Pascal programming language.

An interesting aspect of the project was that the opinions of the experts varied to a greater extent than was expected. The experts had failed to define their solutions and explanations to the accuracy required of expert systems development. Even those solutions agreed upon changed during the development work thus necessitating the continual redefinition of the knowledge base. This was mostly due to new court decisions concerning the application of the collective agreement (the Labour Court, the Supreme Administrative Court).

In order to make the data input as clear and flexible as possible, special attention was paid to the user interface. This was one of the main requirements of the client since the system is intended for users with little computer experience. The number of questions asked during a consultation should be kept to a minimum so that routine cases can be solved quickly and efficiently. For this purpose predefined values, which according to the experts represent the most common cases, were used wherever feasible.

8. Results of the project

8.1 Knowledge representation

It turned out to be impossible to build the expert systems for this project in such a way that they would use solely rule technique. This kind of application necessitates a good deal of calculation for which rule based development is inappropriate. Furthermore, neither is the use of rules conducive to a good user interface. If a program is to be easy to use, some of the rules must be sacrificed. This is not in itself a serious limitation however, since rules are of no value in themselves. If an expert system works satisfactorily with a mixture of rules and algorithmic programming, the use of the conventional programming should not be an obstacle for its acceptance.

One major problem was over how to define the system's boundaries and how to pass on this information to the user via the system itself. In this project the boundaries were defined in the user manual of the program and the systems were built in such a way that they accept input only from the valid area within the boundaries. This solution is not entirely satisfactory in that all information should be contained within the program itself, nevertheless, it is very difficult to build the system 100 % 'watertight' with regard to this matter.

8.2 Knowledge transfer

The building of the one system did not make the development of the other easier to the extent that was first expected. Even though transferring the basic structure proved relatively straightforward, it took a long time to ensure that both systems worked to the same level of efficiency. Initially it was thought that Guru's rules could be transposed directly into the Insight 2+ version. In reality, knowledge transfer was possible only on the knowledge scheme level. The two systems were principally developed by separate persons which, along with the major differences between the shells contributed to the fact that the building of the first system did not help the development of the second. This problem was exacerbated by the need for the Insight 2+ version to carry some extra features (e.g. the final decision on a calendar).

8.3 Knowledge testing

Testing the systems proved quite difficult. Due the many exceptions and complicated rules within the legal domain, it was impossible to order the different cases in a logical matrix form which could then be systematically tested, as is the practice for expert system applications of a more technical domain. No directly applicable methodology for the testing such a kind of expert system was found.

For the purposes of this project, experts from The Federation of the Finnish Building Industry tested the program in those aspects they thought could go wrong and the knowledge engineers then systematically ran many test cases based on the results of the experts' testing.

8.4 Knowledge updating

Since not everything in the expert system can be expressed by rules and as rules become extremely complicated if too much is represented by means of them, it is impossible for the experts to update the system by themselves. The knowledge updating must either be carried out by the knowledge engineer or then the experts must be involved in the development of the systems in order that they may gain a sound inside knowledge of the structure and functioning of the program.

Technical updating and distribution is also an essential question when considering the viability of the micro computer based expert systems. One simple solution, which would not require any complicated technical arrangements, would be simply to send the user new updates on a diskette. Since the collective agreement is renewed only once a year or once every two years this would not normally constitute a problem. A more serious aspect is that However, the solutions and agreed interpretations pose a more serious problem in that they change far more often; merely annual updating would not be sufficient to maintain an accurate and up to date system. Nevertheless, since the current procedure of The Federation of the Finnish Building Industry is to send circular letters to the building companies to inform them of any new interpretations, the problem of diskette updating should be by no means insuperable. Updating via modems might offer one possible solution.

9. Conclusions

It seems that the present expert system technique for micro computers is already suitable for building small knowledge based systems for interpreting the collective agreement and other similar applications in the legal field. Systems covering the whole collective agreement should also be feasible if they are built up of smaller sub-systems.

As better expert system shells are coming to the market so the building of systems becomes easier. This shouldn't however deter one from embarking upon development work now, since the present shells already offer a viable means and systems built now may be transferred to improved shells in the future. Present systems provide a firm basis for further developments.

REFERENCES

1. Hart, A. Knowledge acquisition for expert systems, McGraw-Hill Book Company, New York 1986.

2. Hayes-Roth, F., Waterman, D., Lenat, D. Building Expert Systems, Addison-Wesley, Reading 1983.

3. Holsapple, C. & Whinston, A. Manager's guide to Expert Systems Using GURU, DOW JONES-IRWIN, Homewood 1986.

4. Jolkkonen, E. & Salokivi, J. Sairausajan palkan asiantuntijajärjestelmä, Valokynä April 1987, pp. 30-34.

5. Waterman, D. & Peterson, J. Expert systems for legal decision making, Expert Systems, October 1986, pp. 212-225.