8. KNOWLEDGE BASED SYSTEMS IN MANUFACTURING SIMULATION

8.1 Introduction

8.1.1 Summary introduction

The first part of this section gives a brief overview of some of the different uses of expert systems in manufacturing simulation. Three such uses are described in some detail. The first is the use of an expert system, connected to a simulator, as a teacher. Then we have an expert system, as a part of a simulation environment, doing the traditional scheduling in production management. Finally, the use of knowledge based techniques in the analysis of manufacturing systems are described.

Building simulation models and designing simulation experiments are other areas where expert systems have proven successful [26, 27, 28], but these functions of these systems are not dealt with in this paper. Both Simulation Craft™ [26] and MOSYS [27] also contain expert model analysis, see Section 8.4.

8.1.2 Artificial intelligence, knowledge based systems, expert systems

The goal of artificial intelligence, AI, as a part of computer science, has been to develop computer programs that can solve problems in a way that would be considered intelligent if done by a human [29]. The efforts of AI scientists have been focused on generalised knowledge representation and search techniques for specialised programs:

* Knowledge representation focuses on ways to formulate a problem so that it becomes solvable

* Search focuses on ways to efficiently navigate through the solution space to minimise computation time and memory requirement

Advanced programs that can solve a variety of problems based on stored knowledge, without being reprogrammed, are called knowledge based systems, KBS, [30]. If the program exhibits the same level of problem solving skills as an expert in a specific field, it is called an expert system, ES, [29]. This area of computer science is called knowledge engineering.

The archetypal expert system is MYCIN [31], which is an expert system to guide doctors in diagnosing blood infections. Although this and many other expert systems have proven to be successful, it must be admitted that a not negligible percentage of what is called expert systems is dealing with trivial problems, and should therefore be called knowledge based systems.

Whatever the complexity of the problem the knowledge based system is intended to solve, there are two major tasks to be done, which are certainly not trivial. These tasks are the knowledge acquisition and the knowledge representation. Production management is one of the areas where these tasks can only be done by skilful knowledge engineers. In scheduling and planning at all levels, a lot is done by rules of thumb and "facts" that are difficult to represent using computer languages.

8.1.3 Manufacturing simulation - world views and object orientation

Mitrani [2] describes the three different "world views" of simulation; event-, activity- and process-orientation. I would like to say a few words about the view called process orientation. In this view the real world is modelled as a set of objects, where each modelling object represents an object in the real world. The behaviour of a non-permanent object is described as the process of this object. A process is a sequence of events and activities. These activities require access to the permanent objects, often called resources. An event in manufacturing simulation is typically the start-up and the finishing of a product in a machine.

This object orientation has much in common with what is found as the main feature in many AI programming languages and frames. It has also led to the development of object oriented programming languages as SIMULA [32], and the discrete event simulation extension of SIMULA, the class called DEMOS [4]. The object orientation has also been used by Ulger and Thomasma [33], implementing their simulation control framework for building discrete event simulation models. They used the Smalltalk-80 object oriented programming environment. This paper [33], and also the paper "Towards a Knowledge-based Network Simulation Environment" [34] written by Ruiz-Mier, Talavage, and Ben-Arieh are suggested for further reading.

8.2 Simulation models and expert systems as a teacher

This is the traditional use of expert systems in combination with simulation. The student is communicating with the expert system, answering questions and giving solutions. The system tries to reveal the level of the students knowledge, and to build a model of this knowledge. In some cases the solutions given by the student are so complex that a simulation experiment is needed to check it. The expert system transforms the solution into a set of sub problems, which can be solved by executing an appropriate model on a simulator. The results are presented to the student. The use of graphics is important in the user interface. The typical subject areas are medicine, economics, electronics and ecology.

But it must be admitted that some of these systems are made for use in education in primary school. This means, of course, that the knowledge is not necessarily on expert level.

It is also postulated [35], that the more advanced the problem in concern is, the more important is the simulator. The reason of this is, of course, that in these cases the expert system is dependent of the domain-specific knowledge of the simulator. Remember that the expert system in this context is more an expert on teaching than an expert on a specific domain.

This is also what is the strength of this way of using expert systems together with simulators; the expert systems can use the vast amount of domain-specific and also mathematical expertise that are encoded in the simulators. The most characteristic shortcoming of using simulators is that they have a large number of assumptions built into them.

Other main advantages of these systems are of that this is an interesting, realistic, and not to forget, novel way of learning.

A successful implementation of an expert system using simulation is SOPHIE I, II and III [35]. SOPHIE is an intelligent tutor of electronic-troubleshooting in combination with a general purpose electronic simulator, SPICE.

Simulators are used in teaching of new personnel in today's manufacturing. Used in combination with an expert system as described here, I strongly believe that this will be tomorrow's teaching environment in manufacturing.

In fact we also see EU programs, like COMETT, stimulating the development of such use of simulation software in training.

8.3 Expert systems for scheduling

Many prophecies within the AI field say that using expert systems in scheduling is the most prosperous part of AI in production management. Scheduling can be thought of as finding one good solution from an enormous amount of possible solutions. The result, or the answer from the expert, need not be the optimum solution, it just has to be good enough. This is one of the most developed areas of knowledge engineering. Two of the successful implementations of heuristic schedulers are described in [36, 37].

If an expert system is used in scheduling at any level of production management, a simulation tool used to model this system must have these expert system functions too. But this works the other way round also; it is no use having an expert scheduler in the simulator if the output from the simulation is not applied in production management.

Shivnan and Browne [38] present a method of applying AI based simulation to scheduling. This system, implemented in the rule-based language OPS5, has possibilities of "looking ahead", and predicting what will happen. This means that decisions are made not only considering the current state, but also what happens if this decision is made.

Wy and Wysk [39] present MPECS - An Intelligent Flexible Machining Cell Controller. This system operates in a slightly different way. Deterministic simulation is used to evaluate the scheduling rules that the expert system has come up with.

One of the great advantages of such systems is the sensitivity of minor changes or complications. The effect of a short delay in delivery of raw material, or a breakdown on a machine, can be analysed immediately in a "what-if" manner. But this again, requires a system for updating the expert system with information.

The development of SIMMEK-II, integrated with MRPII system, facilitates the possibility of using simulation and scheduling together. But the simulation and scheduling should be used in a sequence (see Section 7.6), and not as one system.

8.4 Expert systems as an analyst of simulation results

Any simulation run will produce a vast amount of results. The analysis of these results is really the core of a simulation experiment. It is the analysis that tells you that you do not have enough production capacity, that you should apply another scheduling rule, or that you have a bottleneck at the milling machine.

This analysis, which goes beyond the traditional statistical analysis of the results, is by many simulation experts believed to be a promising field for the combination of simulation and knowledge based systems.

A few systems are already implemented or are being developed. These systems are of one of two types; either for analysis of general manufacturing systems, or for analysis of specialised manufacturing systems like flexible manufacturing systems.

One example of the first type is Simulation Craft™ [26]. The expert system of Simulation Craft™ has not only an expert modular and an expert experiment designer, but also an expert model analyser. The expert model analyser, implemented in LISP, does experiment evaluation and alternative generation. It performs a rule-based analysis that incorporates the merged knowledge of experts in the following domains; product costs, purchase, capacity utilisation, scheduler evaluation and shop floor control evaluation.

MOSYS [27] is a tool for the design of FMS using simulation, and is implemented in Prolog. Another example is FASIM/FASDT [40] a system for analysis and design of flexible assembly systems.

Explaining the potential of knowledge based systems in this field, we must again mention these systems' ability to search through a large solution space in an effective way. The search techniques make the expert system capable of incorporating all the information in the model, in the experimental frame and in the results.

8.5 Ideas about an expert system together with SIMMEK

The results from a simulation run are presented to the user on the screen or in hard copies. In the SIMMEK system we have not focused on animation. We have experienced that what production managers really want is the hard figures and numbers. And if the user after studying the results immediately sees what is wrong in the plant, the expert is not consulted. But the amount of results from a simulation of a large and complex system is huge. It is difficult to know which results are essential, and what they really mean.

This is the reason why so many simulation system developers are so keen on building an expert system in combination with the simulation system. In the SIMMEK programme an activity was started to investigate the possibilities of integrating an expert module with the simulation system. Some specifications were made, but it became clear that the resources needed were far beyond what was available in the programme. The main reason for being too costly to implement was the constraint that SIMMEK should be an universal manufacturing simulator. This means that it should cover a large range of different manufacturing systems and companies.

For such a system to be effective it needed to implement not only rules and prescriptions, but also views and opinions. And when these views and opinions are not consistent, not even with themselves, the task is immense.

The specifications that were made, were based on an implementation based on the programming language Prolog, using the in built backtracking search mechanism. Sections 8.5.1 to 8.5.4 will show some of these specifications. Some of the difficulties we ran into concerning knowledge acquisition and knowledge representation are explained in 8.5.5.

8.5.1 The input - The case database

The input to the expert system comes from two sources. The first source is the model structure and model data. Important here is the cost and value factors given in the model building phase. The data from the model will be transformed into facts or unconditional conclusions in Prolog. Some examples are;

```
machine(køping).          highmachinecost(køping).
machine(sharman).                   class_a_product(shaft).
operator(john).           transport(truck,køping,sharman).
product(shaft).           machineoperator(john,sharman).
```

Here we have examples of situations where it is needed to classify objects into categories. To classify a product as an a, b or c product is a well known method. But the classifying of machine cost is not as straightforward. An exact limit, where costs higher than this limit is "high cost", may create unwanted situations. As an example, two machines with almost the same costs, but on each side of the limit, are placed in different categories. But then again, this is the archetypal problem of knowledge based systems; representing inexact rules in a computer.

A similar transformation is performed on the results, with the same difficulty, giving facts like;

```
highmachineutil(sharman).
longthroughputtime(shaft).
```

It is important that these facts are listed in a defined sequence, since Prolog executes the first conclusion first. This means that the most important or at least the mostly used facts should be placed first.

8.5.2 The rule database - Knowledge representation

The knowledge of the expert system is represented as rules, or conditional clauses in Prolog. This has been, and is, the really tricky part of the implementation. The knowledge that is intended to be represented as rules, is going to concern simulation, management, scheduling, technical constraints, economics, and statistics. Some of this knowledge is straightforward to represent, a simplified example is given below;

```
newoperator(X):-machineoperator(Y,X),badmachavail(X),busyoperator(Y).

badmachavail(X):-lowmachineutil(X),longmachinequeue(X).
```

This means that a new operator should be set to operate a machine if the current operator is busy (possibly with other jobs), and the machine availability is bad. This occurs if the machine utilisation is low even though the queue in front of it is long.

This simplified management rule is not so difficult to implement, but this is not true for the majority of the rules concerning simulation, management, scheduling or technical topics. The rules concerning economics and statistics are more exact, and the representation of them does not create as many problems.

Again the sequence of the rules in the database is important, since the first rules are executed first.

8.5.3 The search

The search is performed using the Prolog execution strategy, i.e., top down, left to right, first to last, depth first. The search is initialised by the user with a query. This query is expressed in either a general form, or a more specific one. Examples of the general goals are;

```
        :-reduce_costs(X).
or
        :-increase_throughput(X).
```

A search with one of these goals will search through the case base, trying to solve conditions concerning high costs or low throughput. The performance of these searches will require considerable computer time.

An example of a more specific query is;

```
:-increase_throughput(shaft).
```

The suggestion, i.e., the goal of the search, may be to change some product routing, to rearrange the machine grouping, to put an extra machine, operator, or transport unit into the system, or to change the selection rules. If the modification is approved, it is implemented in the model, and another simulation run is performed. This process may be continued until a satisfying situation is found. It must once more be stressed that the expert does not necessarily find the optimum solution. It goes for what it believes are creating the biggest queues or longest throughput times. It is up to the user to stop the search when a satisfying situation is reached.

8.5.4 The object orientation

The object orientation of the simulation tool simplifies the representation of the facts in the case database. A model in the model database consists of a set of objects. The transformation of the data of these objects is simplified since a sequential search may be used. When for instance a machine object is found, all the data concerning this can be transformed into facts, or unconditional conclusions. All facts about one object can be placed in one section of the case database.

The object orientation is also convenient when model changes are to be implemented. In many cases a modification will only have effect on one object, and thus on a limited part that is easy to identify in the model database.

8.5.5 Knowledge acquisition

As mentioned earlier, knowledge acquisition is always one of the really difficult tasks when building an expert system. We have identified three main sources of knowledge within the area of production management (PM);

1. Books and other written material concerning PM in general

2. Written, company specific instructions, rules and procedures

3. Unwritten instructions, rules and procedures

It is obvious that knowledge acquisition is an easier task when the knowledge is found in writing. This statement is true for production management as it is for economics and statistics. But it is a fact that a considerable amount of the rules applied in production management is not found in writing. And what is even worse; many of these rules are difficult to express in words, since they are rules of thumb, or even instinct.

Apart from studying books and other literature, we have interviewed persons with experience within production management, trying to find the rules mentioned in 3. The next step is to spend some time watching planners and schedulers at work, trying to reveal their "secret" rules, and represent them in our rule database. In this way we hope to be able to create a knowledge based system that really deserves to be called an expert system.

8.5.6 Algorithms for investigating results

The algorithms presented in Sections 6 and 7, connected with the Raufoss case, can be seen as 0-version of a set of algorithms for investigating simulation results. Although the rules put into these algorithms are fairly simple, putting them together is of substantial help in reducing the time needed for each simulation run.

And perhaps this is the area of using KBS like techniques within simulation lies in the nearest future; revealing the human modeler from having to deal with all details from a simulation, and make him or her able to use all effort on the main results.

8.6 Summary remarks on knowledge based systems in manufacturing simulation

This section on knowledge based systems in manufacturing simulation must be seen as a short state of the art, as well as sharing some ideas on use of KBS/AI within SIMMEK. This is very far from what I hoped and aimed for some five years ago.

The main reason for this change in aim and attitude is that during these years of work on simulation in manufacturing, the engineer in me has conquered the idealistic researcher with visions of what a computer tool can do. For a simulation system the rule that if it does not work (and work fast), it is not worth it, is certainly true. And for using simulation at an operational level there are more practical problems than AI problems to solve.

My conclusion is not very promising. I do not see that the future of simulation lies in using AI techniques. There are other research and development areas to examine first.

The most promising area of use of KBS/AI is still in result analysis phase. But may be more in the trivial extraction and presentation, than in the interpretation and model changing parts of it.