KNOWLEDGE BASED IMPROVEMENT: SIMULATION AND AI FOR IMPROVING UNPLANNED MAINTENANCE OPERATIONS

A. Alifantis, S. Robinson*, J.S. Edwards**, R.D. Hurrion*, J. Ladbrook***, T. Waller*****

*	*Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK*
**	*Aston Business School, Aston University, Birmingham, B4 7ET, UK*
***	*Productivity Engineering, Ford Motor Company, Dunton Engineering Centre (15/4A-F04-D), Laindon, Basildon, Essex, SS15 6EE, UK*
****	*Lanner Group, The Oaks, Clews Road, Redditch, Worcs, B98 7ST, UK*

ABSTRACT

Maintenance is one of the largest indirect operating costs for many manufacturing organisations. Meanwhile, many of the decisions relating to maintenance rely upon the knowledge and expertise of a maintenance supervisor. Improvements in the supervisor's decision-making could lead to significant cost savings. Knowledge based improvement is a methodology that uses visual interactive simulation and artificial intelligence in cooperation. It aims to determine what the current decision-making strategies are, and to look for improvements in those strategies. The methodology is being tested on the maintenance operations at a Ford engine assembly plant.

KEY WORDS

Visual Interactive Simulation, Neural Networks, Expert Systems, Unplanned Maintenance

1. INTRODUCTION

Maintenance is rapidly becoming the largest indirect operating cost for manufacturing organisations. A survey of Scandinavian industrial companies found that almost 5% of the companies' turnover was spent on maintenance [12]. Added to this, Wireman [24] found that maintenance costs for US industrial firms rose at a rate of 10%-15% per year during the 1980s, suggesting that this proportion is set to increase. It is apparent that modern 'low cost' manufacturing methods such as lean production, just-in-time and agile manufacturing are contributing to this rise in costs through a greater emphasis on automation and reduced work-in-progress (WIP) making maintenance more critical [8]. Indeed, Mann [14] argues that increased automation leads to manufacturing plant with more workers employed in maintenance than in production.

A key element of the maintenance process is the scheduling maintenance tasks for both unplanned maintenance (equipment failure) and planned maintenance (preventative maintenance and tool-change). This is of critical importance to maintaining manufacturing throughput and minimising WIP. For instance, in a typical Ford engine plant in the order of 3 to 4 engines are lost, or a WIP buffer of 3 to 4 engines is required, for every minute that a machine is not working.

In order to address this issue some have attempted to automate maintenance scheduling through mathematical algorithms [e.g. 21] and heuristic techniques [e.g. 7]. There is, however, only limited practical application of these techniques. In general, maintenance scheduling remains the task of the maintenance supervisor [14].

Reliance on human decision-makers for scheduling decisions has the advantage that the decision-maker often has much experience with the manufacturing facility and is able to cope with many different circumstances. When commissioning new manufacturing facilities, however, the benefits of experience are not present and may take many years to obtain. Added to this are the problems that human decision-makers are unable to manage large quantities of information in short periods of time and are open to making poor and inconsistent decisions. Indeed, research shows that there is great variation in the effectiveness of individual ‘expert’ decision-makers [2]. This has been borne out by the success of expert systems projects intended to bring all decision-makers up to the level of the best available decision-maker, such as Bonissone and Johnson [3]. It is important that maintenance supervisors obtain a knowledge of the process that they are managing and of the effectiveness of their decisions as quickly as possible, as well as identifying alternative strategies for making decisions. In this way the costs of maintenance and WIP, and the loss of throughput from machine down-time can be minimised.

For many years computer simulations have been used to aid the design of manufacturing facilities, reducing the risk of the investment and helping to reduce costs [20]. Simulations have also been used to look at specific aspects of maintenance [e.g. 1, 17]. Their use in evaluating human decision-making within these facilities, however, has been limited because it is difficult to represent the complexities of decision-making within the confines of the software [18]. Meanwhile, artificial intelligence (AI) methods have specifically been developed as a means of representing and improving human decision-making [4, 5, 22]. Therefore, should simulations and AI be linked this could enable the accurate modelling of a human decision-maker [6, 9, 13, 16, 18, 23], which in turn could form the basis of a means for training maintenance supervisors.

The purpose of this paper is to describe a methodology, known as 'Knowledge Based Improvement', that is being developed to help improve the scheduling of maintenance operations. This involves the linking of simulation and AI to provide a learning environment, which in this case is being applied to maintenance operations at the Ford engine assembly plant at Bridgend. The specific focus of the work is on unplanned maintenance. This paper describes the methodology and progress to date in employing it at the Ford plant.

2. THE METHODOLOGY: KNOWLEDGE BASED IMPROVEMENT

The methodology outlined in this paper, known as knowledge based improvement, consists of five key stages:

- Understanding the decision-making process
- Data collection
- Determining the experts’ decision-making strategies
- Determining the consequences of the decision-making strategies
- Seeking improvements

Each of these stages is now described.

2.1 Stage 1: Understanding the Decision-Making Process

The first step in determining the experts' decision-making strategies is to identify the component parts of the decision-making process: decision variables, decision options, attribute variables and attribute levels. For instance, in a simple maintenance scheduling problem where there are 2 actions, that are not mutually exclusive, and 2 engineers who can be asked to act if they are available; as such, there are 4 *decision variables*. The first 2 variables correspond to the actions and the other 2 to the engineers. Each of them has 2 alternative *decision options*: the action can either be taken (denoted 1) or not taken (denoted 0); the engineer can either be asked to act (denoted 1) or not asked to act (denoted 0). Assume, for the simplicity of the example, that the decisions are determined taking into account an estimate of the repair time and the type of fault. It is clear that there are two *attribute variables* in the decision-making process. The first attribute can take the value of any real number that represents the estimated repair time, albeit that in practice it is likely to be rounded to at least the nearest whole number. The second attribute (type of fault) can take values that represent the code of any particular fault. The range of estimated repair times and number of fault codes define the *attribute levels*.

Although interviews and discussion with the decision-makers can reveal some information about the decision-making process, usually the human expert cannot explicitly identify and list the decision-making components. To do so the modeller should observe the human experts as they take decisions. In addition, in order to build a complete model of the decision-making process the modeller may need to make assumptions by considering other rational decision that can be taken by the decision-maker.

A decision-making process can be represented as two row vectors. The first vector corresponds to the decision, with each element representing a decision variable (d). The second vector corresponds to the attributes of the decision, with each element representing an attribute variable (a). In the context of the simple maintenance scheduling example described above, the decision-making process can be represented as follows:

$$\mathbf{D}_{i,j} = f_j(\mathbf{A}_i) \qquad (1)$$

Where:

$$\begin{aligned} \mathbf{D}_{i,j} &= [d_1 \quad d_2] \\ \mathbf{A}_i &= [a_1 \quad a_2] \end{aligned} \qquad (2)$$

The subscript i indicates the time at which the decision was taken and the subscript j indicates the human expert that took the decision. The function f represents the decision-making strategy of the individual expert, taking into account the attributes of the vector $\mathbf{A_i}$. The purpose of stages 1 to 3 of the methodology is to determine the function f by applying AI techniques to a set of collected example decisions.

2.2 Stage 2: Data Collection

Having identified the decision components, the next step in determining the decision-making strategies is to collect examples of decisions from each expert. Each example in the data set should include the value of each decision and attribute variable. The data set should have the form of two matrices: $\mathbf{D_j}$ and $\mathbf{A}$. $\mathbf{D_j}$ represents the decisions made by decision-maker j under specific attribute values (identified in $\mathbf{A}$). Each row of the matrix $\mathbf{D_j}$ corresponds to the row vector $\mathbf{D_{i,j}}$, that is, the decisions taken at time i. Each column in the matrix $\mathbf{D}$ corresponds to a decision variable. Each row of the matrix $\mathbf{A}$ includes the attribute values at a particular decision point (i). Each column corresponds to an attribute variable.

For the simple decision-making process outlined above, the data set to be used in determining the decision-making strategy of expert j should have the following form:

$$\mathbf{D}_j = \begin{bmatrix} d_{1,1} & d_{1,2} \\ . & . \\ d_{i,1} & d_{i,2} \\ . & . \\ d_{I,1} & d_{I,1} \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ . & . \\ a_{i,1} & a_{i,2} \\ . & . \\ a_{I,1} & a_{I,1} \end{bmatrix} \qquad (3)$$

One method of collecting these data would be through observation of the experts at work. This, however, would be extremely time consuming, particularly if the elapsed time between decision points is large. It would also be difficult to record the full set of many attribute values at a specific moment in time, and because the values are likely to change continuously, inaccuracies would occur if there were any delay. As a result, the methodology uses a VIS. The expert interacts with a visual simulation of the system in question. The simulation model stops at a decision point and reports the values of the attribute variables. The expert is then prompted to enter his/her decision to the model. The model records the value of each decision variable and attribute variable to a data file. As a result a set of values for the matrices $\mathbf{D_j}$ and $\mathbf{A}$ are collected.

The methodology suggests the use of VIS for a number of reasons. First, it is less time consuming than observation, because the simulation runs much faster than real time. Second, because the simulation stops at a decision point, it is possible to capture all attribute values at that moment in time. A third benefit is that a simulation run can be replicated exactly, enabling the system state to be interrogated further at a later date, should this be required. For instance, it may become apparent that the decision-maker takes into account attributes that have not previously been identified. This also provides the benefit that different decision-makers can be presented with the same series of attribute values.

Of course the use of VIS as a data collection method is not free of problems. Three specific difficulties arise. First, the model needs to contain and report all the key attributes in the decision-making process. This probably requires a very detailed model which in itself could be time consuming to develop. Accurate data, required to support such a detailed model, may not be available either. A second problem is the need to involve the human decision-maker in entering decisions to the model. A very large number of example decisions may be required to obtain a full set of data, which in itself could be time consuming. A third problem is whether the human decision-makers are likely to take realistic decisions in a simulated environment. It is quite likely that they will take greater risks, as there are no real consequences to their decisions.

Anticipating the second problem, the methodology suggests the collection of a limited number of decision examples using the VIS. This data set is not expected to be large enough to be able to determine the decision-making strategy of an expert. It is believed, however, that it could be used to train a neural network model. This in turn could be used to increase the example set $\mathbf{D_j}$ and **A**. A feed forward neural network with 3 layers is believed to be computationally sufficient for this purpose. A separate neural network would, of course, need to be trained for each decision-maker.

2.3 Stage 3: Determining the Experts' Decision-Making Strategies

Having collected a series of examples using the VIS and the neural network model, the next step is to use the data in the matrices $\mathbf{D_j}$ and **A** to determine the decision-making strategies of the individual experts. A decision-making strategy can be represented by the use of a decision tree; a separate decision tree being constructed for each decision-maker. Experts systems software is capable of constructing a decision tree from a set of examples, such as those collected via the VIS and neural network. One such method for constructing a decision tree is Quinlan's ID3 algorithm; see, for example, Mingers [15]. The algorithm prioritises the attributes according to the degree to which they match the data set with the correct decisions.

2.4 Stage 4: Determining the Consequences of the Decision-Making Strategies

Having determined the decision-making strategies, that is, a decision model f_j for each expert j, the next step in the knowledge based improvement methodology is to assess and compare the performance of each expert. The ultimate performance measure in most manufacturing facilities is the level of throughput. This means that each expert can be assessed on the basis of the throughput that is achieved in the simulation when the decision-making process is controlled using his/her decision-making strategy. To predict the throughput, conditional to each human expert, the VIS can be linked with the expert systems software (or, indeed, the neural network)

The expert system is used in place of a decision-maker to interact with the simulation. Each time the simulation reaches a decision point the simulation stops and the expert systems software is invoked. The value of each decision attribute is passed from the simulation to the expert system software. In turn, the expert system returns the values of the decision variables to the simulation before the simulation run is continued. For a description of how to link a simulation package with expert systems software see Robinson et al [19].

When the simulation has reached the end of the run, the throughput of the production line provides an indicator of the performance of the expert whose decision tree was used during the run. Running the simulation under each expert's decision-making strategy for a number of replications to eliminate stochastic variability, enables the most efficient strategy to be found by comparing the output from each run.

Of course having identified the most efficient expert does not mean that the most efficient strategy has been found since there is no guarantee that the best current strategy is the optimal one. Although the best strategy may not be optimal it can still be used to train less efficient decision-makers, providing improvements in overall performance.

2.5 Stage 5: Seeking Improvements

The last stage in the methodology uses the decision-making strategies of the most efficient experts as starting point to search for an improved strategy. The search could be made informally by combining strategies and by making incremental changes. Alternatively, heuristic search methods could be implemented, in order to seek for improvements. In each case, the alternative strategies can be tested by running them with the simulation model in order to determine their effectiveness.

3. CASE STUDY: FORD BRIDGEND ENGINE ASSEMBLY PLANT

3.1 Background to the Case Study

The Ford engine plant at Bridgend is one of the main production facilities for the 'Zetec' petrol engine. The plant consists of a number of transfer lines [10] that feed the main engine assembly line. In engine assembly, blocks are placed on a 'platten' and pass through a series of automated and manual processes. For the purposes of this research, the maintenance operations on a self-contained section of the engine assembly line are considered.

Prior to this research one of the authors (Ladbrook) had already developed a simulation model of the complete engine assembly facility. The model, developed in the WITNESS simulation software [11], was used to identify bottlenecks and to determine viable operating alternatives. The maintenance logic in the model assumed that when a machine fault occurred, the decision would be to make an immediate repair. Random sampling was used to determine the skill level of the engineer required to service the fault. These assumptions were considered to be adequate for the purposes of the study that was performed. No study has been undertaken, however, to evaluate the effectiveness of alternative maintenance strategies.

In the early stages of this research one of the authors (Alifantis) spent some time observing the production facility and in discussion with plant engineers. This showed that reality is somewhat different from what is represented within the simulation. Although the obvious action to take when a machine breaks down is to repair it immediately (RI) this may not always be the most appropriate action for a variety of reasons:

- *Inappropriate*: If there is a long queue of parts downstream from the machine requiring repair, then immediate repair may not be the most appropriate action, and the maintenance engineers may be better deployed elsewhere.
- *Insufficient*: Repairing a machine takes time. Meanwhile the rest of the production facility continues to process parts and to move them around. This means that during the repair of the

machine queues may occur upstream, while downstream the process will be starved of parts. Simply repairing the machine may be insufficient to reach target throughput.

- *Impossible*: Sometimes it may not be possible to repair the machine immediately since all the maintenance engineers are busy. There is always the option to interrupt the repair of another machine and so to release one of the engineers, but this may not be the best course of action. Further to this, on occasions spare parts required for the repair of the machine may not be available.

From the above it is clear that apart from repairing immediately other policies should be considered when a machine breaks down.

Stand-by (SB) can be considered as an alternative to repair immediately. In this case an engineer processes the parts manually and pushes them to the next machine through the conveyor. In general it is not possible to repair the machine at the same time as stand-by is being operated because of space restrictions. The type of fault, the extent of queues and labour availability among other attributes are the key determinants of this decision.

Stop the line (SL) is another option, which might be considered as a complement or substitute to repair immediately. In this case the maintenance supervisor ('group leader') should decide whether it is useful to stop the whole line or part of it. This might be used, for instance, to avoid a build up of work-in-progress in a section of the line.

Do nothing (DN) is an alternative decision that might be the desired course of action under certain circumstances, for instance, close to the end of a shift. Obviously this decision must be revised eventually and the machine repaired later.

Beyond the above list of options, the group leader may want to revise the decision for a particular machine at a later point. For example, the group leader might decide to stop the repair of a machine because he needs the labour to repair another machine that is down.

Although it is possible to identify the types of decisions the group leaders might make, it is another to determine how those decisions are taken. In determining what course of action to take the group leaders rely upon their knowledge and experience (tacit knowledge). Direct questioning of group leaders showed they are unable to directly express this knowledge. The knowledge based improvement approach aims to overcome this problem by using the interaction with the VIS as a method of knowledge elicitation.

3.2 Implementing the Knowledge Based Improvement Methodology at Ford

Having provided the background to the case study, the progress in applying the knowledge based improvement methodology is now described. As already stated, the methodology is being implemented by considering a self-contained section of the production facility. In this section a team of 5 engineers (2 skilled, 2 semi-skilled and 1 unskilled engineer) perform unplanned maintenance, among other tasks. One of the skilled engineers acts as group leader for that part of the line. One of his duties is to decide what to do when a machine breaks down. When a failure occurs a message is received via a pager that is carried at all times. The message reports the name of the equipment and a short description of the fault. After an inspection of the machine the group leader decides what action is the most appropriate. The main options available are the following:

SB:	Stand-by
RI:	Repair immediately
SMLB:	Stop the main line before the machine
SMLA:	Stop the main line after the machine
SSB:	Stop the section before the machine
SSA:	Stop the section after the machine

Alternatively, the group leader may decide to do nothing, which is equivalent to a null response to all these options.

Having decided what course of action to take, the group leader should decide who must act. Available engineers who can be asked to act if they are available are the following:

L1:	Group leader
L2:	Second skilled engineer ('M/Elec')
L3:	Semiskilled engineer 1 ('IMS1')
L4:	Semiskilled engineer 2 ('IMS2')
L5:	Unskilled engineer.

Given the above discussion the row vector $\mathbf{D_{i,j}}$ in this particular decision-making process should include the following elements:

SB	RI	SMLB	SMLA	SSB	SSA	L1	L2	L3	L4	L5

Each element of the above matrix represents a decision variable that can take the value 0 or 1. Zero means that the decision-maker is not taking the particular action or that the particular resource will not be asked to act. On the contrary, 1 means that the decision-maker is taking the particular action or he has decided to ask the particular engineer to act. So, for example, if the group leader (j=1) in his first decision decides to repair immediately, and he also decides that the person who should do this is the second skilled engineer, then the row vector $\mathbf{D_{1,1}}$ should be the following:

$$\mathbf{D}_{1,1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad (4)$$

Having identified the decision variables the next step was to find out which attributes are taken into account when making a decision. According to discussion with the group leader, and observation of working practices, the following attributes are taken into account when making a decision:

M:	Machine number
TF:	Type of fault
NPP:	Number of parts produced this shift

T:	Time
ERT:	Estimated repair time
ASP:	Availability of spare part (0 – available, 1 – not available)
NOMD:	Number of other machines down
LA1…LA5:	Labour availability engineer1…engineer5 (0 – available, 1 –not available)
MA1…MAk:	Status of machines (0 – available, 1 – broken down)

This means that the attribute vector $\mathbf{A_i}$ will have the following columns:

	M	**TF**	**NPP**	**T**	**ERT**	**ASP**	**NOMD**	**LA*1*… LA*5***	**MA*1*…MA*k***
Type	**Integer**	**Integer**	**Integer**	**Real**	**Real**	**0/1**	**Integer**	**0/1**	**0/1**

For example, assuming that at the first decision point the system has the following attribute values:

- Machine 1130 has broken down,
- The fault type code is 300
- The number of parts produced this shift is 549
- The time is 13:23
- The estimated repair time is 5 minutes
- The required spare part is available
- No other machines are down
- All engineers are available
- All machines are available

The value of each element of the attribute vector $\mathbf{A_1}$ should be the following:

$$\mathbf{A}_1 = \begin{bmatrix} 1130 & 300 & 549 & 13.23 & 5.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \end{bmatrix} \tag{5}$$

Having identified the components of the decision-making process, the next step is to collect example decisions from the decision-makers via the simulation model. As already stated, the original simulation model assumed that when a machine breaks down the decision is to repair it immediately. As a result, the model required modification to enable alternative decisions to be entered and implemented within the simulation. The new version of the simulation takes into account the decision-maker and stops when a decision point is reached, that is, when a machine breaks down. A Visual Basic front end is invoked and informs the user about the state of the system by reporting all the relevant attributes. The front end is shown in figure 1. The top part of the window provides information on the decision attributes, the lower part asks for input regarding the decisions that are to be taken. The model user can also view the status of the model via the visual display of the production facility.

Once the user has made his/her decision, the simulation run continues by pressing the proceed button. The simulation model then continues to run with the decision implemented within the model. For example, if the decision was to stop the line before the machine, the simulation continues but it does not process the parts which are in the part of the line before the broken machine. The simulation is capable of recording the attribute row vector $\mathbf{A_i}$ and the decision matrix $\mathbf{D_{i,j}}$ in a data file as the model runs.

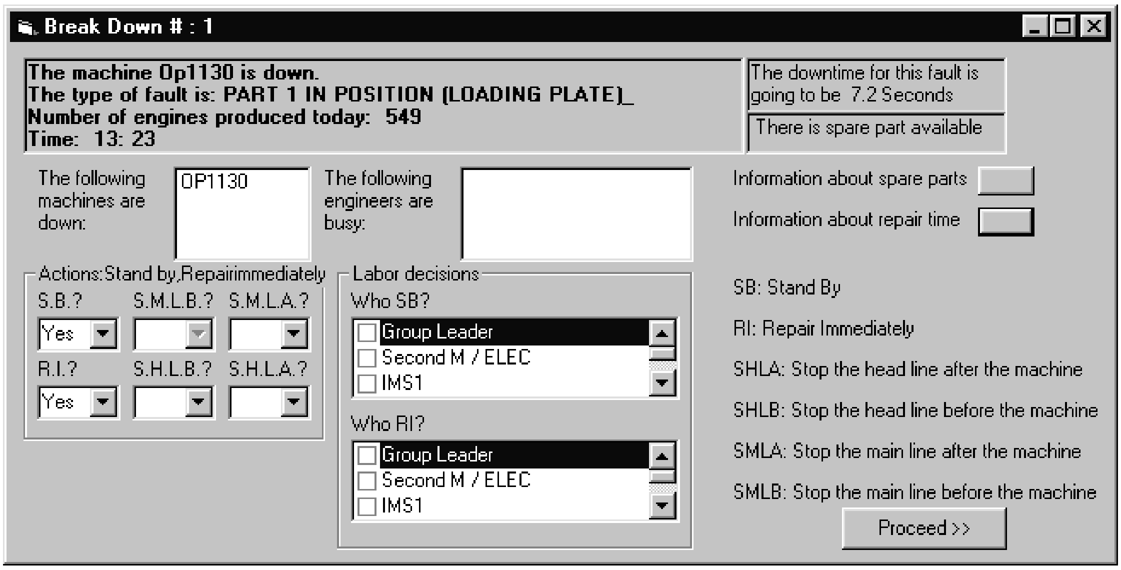

Figure 1 *Visual Basic Front End for the Ford Engine Assembly Model*

To date the model has been used for initial data collection sessions. In these ‘test sessions’ a number of problems were identified. For instance, in the first data collection session the human expert found that the model does not report extremely important attributes such as the estimated repair time for each fault. In the second session it was found that the breakdown scenarios reported were very similar (the estimated repair time for all of them was very short), so the decision by the human expert was the same at every decision point (repair immediately). As a result of these findings, improvements have been made to the model: the estimated repair time is now reported, and the trace data on breakdowns has been adjusted to provide a wide range of scenarios. The next step is to collect example decisions from a number of experts before moving to the AI stage in the methodology.

4. FINDINGS AND DISCUSSION

Having almost completed the stage in the methodology that requires the use of a VIS for collecting example decisions, a number of strengths and weakness in the methodology have been identified.

First of all it has been found that VIS is probably a unique way of obtaining data about decisions in a reasonable time interval, since the simulation runs faster than real time. In addition, VIS is a very efficient approach for data collection since it is an experimental environment where the modeller can control the values of the attributes that are generated and reported to the user of the model. This means that the modeller can guarantee that a wide range of attribute value combinations can be created during a data collection exercise. There is no such control in a real life data collection exercise.

Although the approach is quite promising, some problems have been identified in its application. First of all, it is difficult to isolate and understand the decision-making surrounding unplanned maintenance since the team performs many other tasks as well. Indeed, the team find it difficult to articulate how they make decisions concerning only this part of their work. Another challenging point is the fact that the simulation model may not be capable of reporting all the attributes that the decision-maker takes into account. In the above application, for example, the decision-maker takes into account the physical condition of the machine. This attribute is all but impossible to simulate in a model of this nature.

5. CONCLUSION

The paper has described a methodology that is to be used to evaluate the effectiveness of decision-makers. In addition it proposes a method to improve the current decision-making practices based on the experts' extant knowledge. The methodology described in the paper includes an innovative data collection method based on the use of a VIS combined with a neural network algorithm.

The methodology is being tested with unplanned maintenance operations in a Ford engine assembly line. Initial data on example decisions has been collected via a simulation model. The next stage in the research is to collect a full set of data for different maintenance supervisors. Following this, the use of AI for learning and improving current decision-making strategies is to be explored.

ACKNOWLEDGEMENT

This work is being jointly funded by the EPSRC (grant reference GR/M72876), Ford Motor Company and the Lanner Group.

REFERENCES

1. Al-Zubaidi, H. and Christer, A.H. (1997). Maintenance Manpower Modelling for a Hospital Building Complex. *Eur J Opl Res*, **99** (3), pp. 603-618.
2. Augustine, N.R. (1979). Distribution of Expertise. *Defense Systems Management,* Spring, pp. 50ff.
3. Bonissone, P.P. and Johnson, H.E. Jr. (1984). Expert Systems for Diesel Electric Locomotive Repair. *Human Systems Management,* **4** (4), pp. 255-262.
4. Edwards, J.S. (1991). *Building Knowledge-Based Systems: Towards A Methodology*. Pitman, London.
5. Feigenbaum, E., Nii, H.P. and McCorduck, P. (1988). *The Rise of the Expert Company*. Times Books, NY.
6. Flitman A.M. and Hurrion, R.D. (1987). Linking Discrete-Event Simulation Models with Expert Systems. *J. Opl Res. Soc.*, **38** (8), pp. 723-734.

7. Gopalakrishnan, M., Ahire, S.L. and Miller, D.M. (1997). Maximizing the Effectiveness of a Preventive Maintenance System: An Adaptive Modeling Approach. *Mgmt Sci.*, **43** (6), pp. 827-840.
8. Hancock, W.M. and Zayko, M.J. (1998). Lean Production: Implementation Problems. *IIE Solutions*, **30** (6), pp. 38-42.
9. Hurrion, R.D. (1991). Intelligent Visual Interactive Modelling. *Eur J Opl Res*, **54** (3), pp. 349-356.
10. Ladbrook, J. (1998). Modeling Breakdowns: An Inquest. MPhil Thesis, University of Birmingham.
11. Lanner Group (2000). WITNESS 2000 User Manual. The Oaks, Clews Road, Redditch, UK.
12. Luxhoj, J.T., Riis, J.O. and Thorsteinsson, U. (1997). Trends and Perspectives in Industrial Maintenance Management. *Journal of Manufacturing Systems*, **16** (6), pp. 437-453.
13. Lyu, J. and Gunasekaran A. (1997). An Intelligent Simulation Model to Evaluate Scheduling Strategies in a Steel Company. *International Journal of Systems Science*, **28** (6), pp. 611-616.
14. Mann, L. (1983). *Maintenance Management, 2nd ed.* D.C. Heath, Lexington, MA.
15. Mingers, J. (1986). Expert Systems: Experiments with Rule Induction. *J. Opl Res. Soc.*, **37** (11), pp. 1031-1037.
16. O'Keefe, R.M. (1989). The Role of Artificial Intelligence in Discrete-Event Simulation. *Artificial Intelligence, Simulation and Modeling* (L. E. Widman, K.A. Loparo and N.R. Neilsen, eds.), pp. 359-379. Wiley, NY.
17. Robinson, S. (1993). The Application of Computer Simulation in Manufacturing. *Integrated Manufacturing Systems*, **4** (4), pp. 18-23.
18. Robinson, S., Edwards, J.S. and Yongfa, W. (1998). An Expert Systems Approach to Simulating the Human Decision Maker. *Winter Simulation Conference 1998* (D.J. Medeiros, E.F. Watson, M. Manivannan, J. Carson, eds.), The Society for Computer Simulation, San Diego, CA, pp. 1541-1545.
19. Robinson, S., Edwards, J.S. and Yongfa, W. (2001). Linking Witness to an Expert System to Represent a Decision-Making Process. *Simulation*. In review.
20. Simulation Study Group (1991). Simulation in UK Manufacturing Industry. DTI Study Report (Horrocks, R., ed.). The Management Consulting Group, University of Warwick, UK.
21. Taylor, R.W. (1996). A Linear Programming Model to Manage the Maintenance Backlog. *Omega*, **24** (2), pp. 217-227.
22. Waterman, D.A. (1986). *A Guide to Expert Systems.* Addison-Wesley, Reading, Mass.
23. Williams, T. (1996). Simulating the Man-in-the-Loop. *OR Insight*, **9** (4), pp. 17-21.
24. Wireman, T. (1990). *World Class Maintenance Management*. Industrial Press.