Knowledge Based Improvement: Simulation and Artificial Intelligence for Identifying and Improving Human Decision-Making in an Operations System

Stewart Robinson[1], Thanos Alifantis[1], John S. Edwards[2],
John Ladbrook[3] and Anthony Waller[4]

1. Operational Research and Systems Group, Warwick Business School, University of Warwick, Coventry, CV4 7AL, United Kingdom

2. Operations and Information Management Group, Aston Business School, Aston University, Birmingham, B4 7ET, United Kingdom

3. Productivity Engineering, Manufacturing & Plant Engineering: Inline Gasoline & Diesel Engines, Ford Motor Company, Dunton Engineering Centre (15/4A-F04-D), Laindon, Basildon, Essex, SS15 6EE, United Kingdom

4. Lanner Group, The Oaks, Clews Road, Redditch, Worcestershire, B98 7ST, United Kingdom

* Corresponding author

Abstract

The performance of most operations systems is significantly affected by the interaction of human decision-makers. A methodology, based on the use of visual interactive simulation (VIS) and artificial intelligence (AI), is described that aims to identify and improve human decision-making in operations systems. The methodology, known as 'knowledge based improvement' (KBI), elicits knowledge from a decision-maker via a VIS and then uses AI methods to represent the decision-making. By linking the VIS and AI representation it is possible to predict the performance of the operations system under different decision-making strategies and to search for improved strategies. The KBI methodology is applied to the decision-making surrounding unplanned maintenance operations at a Ford Motor Company engine assembly plant.

Key Words

Simulation, Artificial Intelligence, Human Decision-Making, Knowledge Elicitation, Expert System

Knowledge Based Improvement: Simulation and Artificial Intelligence for Identifying and Improving Human Decision-Making in an Operations System

Introduction

Most operations systems include significant elements of human decision-making and interaction. A manufacturing plant may involve any number of manual processes and many aspects of the plant's operation such as scheduling of production and maintenance operations may involve human decision-making. The situation is normally more exaggerated in service operations where service personnel interact with customers. Meanwhile, these decisions and interaction may have a great impact on the performance of the operations system [Baines and Kay, 2002].

Simulation is often used to model operations systems, but normally this either overlooks the effects of human decision-making or adopts a very simplistic approach. This is probably because modelling these elements of an operations system presents a number of challenges. Most simulation tools do not provide a full set of functions that would be useful for modelling human decisions. A more fundamental challenge is determining the decision-making strategies adopted by individuals within the operations system. Further to this, the purpose of the simulation exercise may be to establish if the human decision-making process could be improved. This requires a methodology for determining, modelling and looking for improvements in human interaction.

The purpose of this paper is to outline a methodology aimed at identifying and improving human decision-making within operations systems. The methodology, known as 'knowledge based improvement' (KBI), is based on the use of visual interactive simulation (VIS) with artificial intelligence (AI). KBI is tested on an engine assembly line at a Ford Motor Company plant in Wales, by investigating the decision-making surrounding unplanned maintenance operations.

The paper starts with a brief review of previous work in which simulation and AI have been linked for various purposes, including the representation of human decision-making. The KBI methodology is then described. Following this, the background to the case study is outlined before describing in detail the application of the methodology. The paper concludes

with a discussion on some issues surrounding the KBI methodology and continuing research work.

Artificial Intelligence in the Life-Cycle of Simulation Studies

It has been proposed that AI could aid the development and use of simulations throughout the life-cycle of a simulation study [1]. Indeed, there are examples of AI being applied at every stage of a simulation study, from model conception to experimentation and the analysis of results. An early attempt at automating the development of conceptual models can be found in Doukidis and Paul [2]. Later, however, it is conceded that intelligent front ends probably provide a less rigid and, therefore, more useful approach [1(XXX renumber)].

Input data modelling provides a role for AI in the simulation life-cycle. Hurrion[3] trains a neural network with an empirical distribution and proposes that the approach might be used to generate random variates for a simulation model.

In terms of model development, there have been attempts at using AI to automatically generate simulation program code, for instance, CASM [4] and Mathewson [5]. AI have also been used for model verification and validation. Doukidis [6] uses an expert system, SIPDES, to help locate and resolve compilation errors in simulation programs. Deslanders and Pierreval [7] develop a system with limited capability for aiding model validation.

As an aid to experimentation and results analysis, there is considerable scope for applying AI. For instance, Hurrion [8] uses an expert system to aid the design of experiments. He also employs a neural network to analyse a simulation model's output [9, 10] [Hurrion, 2000] and as a basis for simulation optimisation [11]. AI can also be used to select experimental scenarios. Pierreval [1992], for instance, uses an expert system to select priority rules for a flexible manufacturing system, testing the rules through a simulation.

Some have used AI as a means for representing human decision-making in simulations[12,13,14,15] [Moffat, 2000]. AI, representing a human decision-maker, is linked to a simulation model. When a point in the simulation run is reached at which a decision needs to be taken, the AI is invoked. The resulting decision is then implemented in the simulation model and the run continued.

It would seem that the presence of human decision-making within simulation models presents two problems to the simulation modeller. First, it is necessary to determine the way in which the decisions are made by the people involved. This is not straightforward, as decision-makers are not always able to articulate their decision-making process. Second, simulation software does not provide a natural environment for expressing decision-making logic. Most of the work in this area has focused on the second problem by linking a simulation with AI. There has been only limited attention paid to the former issue. Indeed, neither is there much concentration in the AI literature on how examples are obtained and used in the process of knowledge elicitation [Kidd, 1987]. This paper attempts to redress this balance by describing a methodology for eliciting knowledge and improving decision-making, applying the concepts to an industrial setting.

Knowledge Based Improvement (KBI)

The prime motivation of KBI is to elicit knowledge on how decision-makers make decisions in an operations system (*decision-making strategies*) and to identify ways of improving their decision-making. The motivation is not to develop a more detailed simulation model, albeit that this is one outcome of the methodology. More detailed models are not necessarily better models [Pidd, 2003] nor are they always more accurate [Robinson, 1994]. The increase in detail required for KBI is aimed firstly at knowledge elicitation and secondly at being able to represent the consequences of alternative decision-making strategies. Neither is the motivation to develop an AI system that replaces the decision-makers in the operations system. It is recognised that the simulation from which knowledge is elicited is a simplified representation of the real system. The full complexity of the real system and the decision-making scenarios is not (and cannot) be represented. Therefore, the knowledge elicited is imperfect. As such it would be a mistake to assume that any AI system developed from the methodology could fully replace a decision-maker. Instead, the aim is to facilitate improved decision-making by identifying improvements to the decision-making strategies employed.
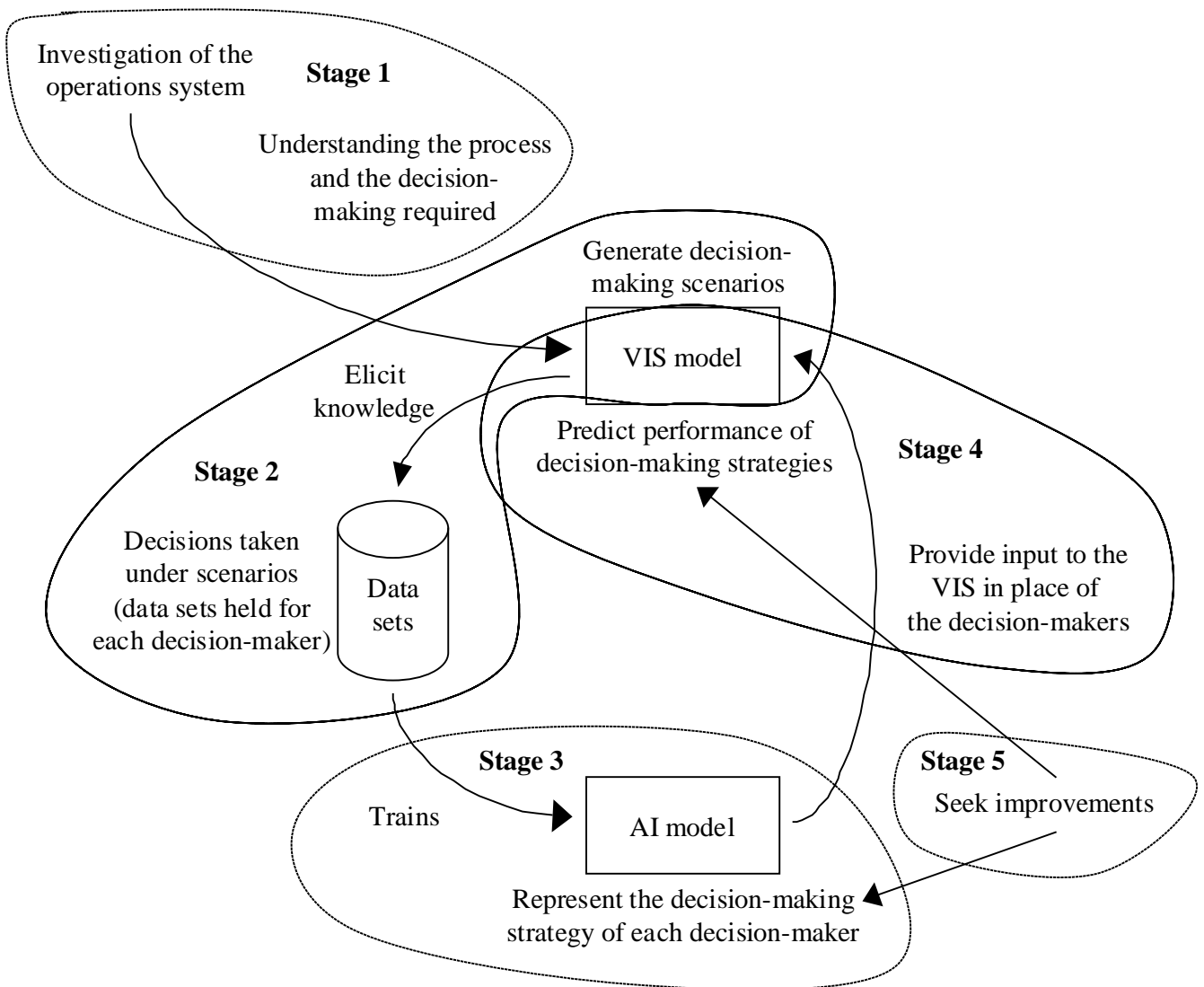
The KBI methodology is outlined in figure 1 and consists of five stages:

- Understanding the decision-making process

- Data collection
- Determining the decision-makers' decision-making strategies
- Determining the consequences of the decision-making strategies
- Seeking improvements

Although the stages are generally performed in a linear manner, there is some level of iteration between them. Each of the stages is now described in some detail.

Figure 1 *Knowledge Based Improvement*

Investigation of the operations system

Stage 1

Understanding the process and the decision-making required

Generate decision-making scenarios

VIS model

Elicit knowledge

Predict performance of decision-making strategies

Stage 4

Stage 2

Decisions taken under scenarios (data sets held for each decision-maker)

Data sets

Provide input to the VIS in place of the decision-makers

Stage 3

Trains

AI model

Stage 5

Seek improvements

Represent the decision-making strategy of each decision-maker

Stage 1: Understanding the Decision-Making Process

The first step in determining the decision-makers strategies is to identify the component parts of the decision-making process: decision variables, decision options, decision attributes and attribute levels. For instance, in the case of a machine failure on a manufacturing line, the decision-maker might have two decisions to make (*decision variables)*, when to repair the machine and who to ask to repair the machine. For each decision variable there might be two *decision options*: the machine might be repaired immediately or left until the end of the shift; an operator might be asked to repair the machine or a skilled engineer. 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 *decision attributes* in the decision-making process. The range of estimated repair times and the number of fault types define the *attribute levels*.

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 a decision attribute (*a*). In the context of the simple maintenance scheduling example described above, the decision-making process can be represented as follows:

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

where:

$$\mathbf{D}_{i,j} = \begin{bmatrix} d_1 & d_2 \end{bmatrix}$$
$$\mathbf{A}_i = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$$

The subscript *i* indicates the sequence in which the decision was taken and the subscript *j* indicates the decision-maker that took the decision. The function $f_j(\mathbf{A}_i)$ represents the decision-making strategy of the individual decision-maker, 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 $f_j(\mathbf{A}_i)$ by applying AI techniques to a set of collected example decisions.

The aim of the first stage of the methodology is to identify the decision variables, decision options, decision attributes and attribute levels. Although interviews and discussion with the decision-makers can reveal some information about the decision-making process, usually the decision-maker cannot explicitly identify and list the decision-making components. To do so the modeller should observe the human decision-makers 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 decisions that can be taken by the decision-maker and attributes that might be considered (hypothesizing). Later in the methodology the decision-makers are asked to interact with the simulation model. This too can reveal further information on the variables and attributes of the decision-making process.

Stage 2: Data Collection

Having identified the component parts of the decision-making process, the next step is to collect example decisions from the decision-makers. Each example in the data set should include the value of each decision option and attribute level. The data takes 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 levels (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 decision point i. Each column in the matrix $\mathbf{D}_j$ corresponds to a decision variable. Each row of the matrix $\mathbf{A}$ includes the attribute levels at a particular decision point (i). Each column corresponds to a decision attribute.

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

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

One method of collecting these data would be through observation of the decision-makers 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 decision-maker interacts with a visual simulation of the system in question. The simulation model stops at a decision point and reports the values of the relevant decision attributes. The decision-maker is then prompted to enter his/her decision to the model. The model records the value of each decision variable and decision attribute to a data file. As a result a set of values for the matrices $\mathbf{D}_j$ and $\mathbf{A}$ are collected. Separate $\mathbf{D}_j$ matrices can be generated for each decision-maker by presenting them with the same set of scenarios $\mathbf{A}$.

Stage 3: Determining the Decision-Makers' Decision-Making Strategies

Having collected a series of examples using the VIS, the next step is to use the data in the matrices $\mathbf{D}_j$ and $\mathbf{A}$ to determine the decision-making strategies of the individual decision-makers. Various AI approaches might be used to determine the decision-making strategies from the example decisions, for instance, rule-based expert systems, regression analysis and artificial neural networks. Rule based systems, that provide a decision tree, are favoured, since they not only represent a decision-making strategy, but they also have greater explanatory power. Expert systems software is capable of constructing a decision tree from a set of examples. One such method for constructing a decision tree is Quinlan's ID3 algorithm; see, for example, Mingers (1986). The algorithm prioritises the attributes according to the degree to which they match the data set with the correct decisions. A separate decision tree needs to be derived for each decision-maker.

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

Having determined the decision-making strategies, that is, a decision model $f_j(\mathbf{A}_i)$ for each decision-maker j, the next step in KBI is to assess and compare the performance of each decision-maker. A decision-maker's performance can be assessed on the basis of the simulation result when the model is controlled by his/her decision-making strategy. This can be achieved by linking the AI representation of a decision-maker with the VIS [Robinson et al, 2003]. The AI representation 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 AI representation is invoked. The value of each decision attribute is passed from the simulation to the AI representation. In turn, the AI representation returns the values of the decision variables to the simulation before the simulation run is continued.

The best strategy can be identified by comparing the results of the simulation when it is run under each decision-making strategy. Of course the strategy identified as 'best' is simply the best from among those decision-makers from whom knowledge has been elicited. Other decision-makers may employ better strategies and the 'best' strategy could probably be improved upon, which is the purpose of the final stage of the methodology.

Stage 5: Seeking Improvements

The last stage in the methodology uses the identified decision-making strategies, particularly of the best decision-makers, as a starting point to search for improved strategies. The search can be made informally by discussion with the decision-makers and by combining strategies and 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 VIS in order to determine their effectiveness.

Case Study: Unplanned Maintenance at a Ford Engine Assembly Plant

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 (Ladbrook, 1998) 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. From time-to-time automated machines breakdown and require repair (unplanned maintenance). It is the decisions surrounding what happens when a machine fails that are the focus of this research. The plant works to three shifts over five days. In order to limit the scale of the work, only the first section of the line was investigated. This involves about a quarter of the total operations in the engine assembly process.

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 [Lanner Group, 2003], 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.

Application of the KBI Methodology to the Ford Engine Plant

Stage 1: Understanding the Decision-Making Process

As already stated, the KBI methodology has been implemented by considering only the first section of the engine assembly line. In this section a team of five engineers (two skilled, two semi-skilled and one unskilled engineer) perform the unplanned maintenance, as well as a number of 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. It is at this point that the original simulation model assumed the machine would be repaired immediately. The reality, however, is quite different.

In the early stages of the research one of the authors (Alifantis) spent some time observing the production facility and in discussion with the plant engineers. Although the obvious action to take when a machine breaks down is to repair it immediately this may not always be the most appropriate action for a variety of reasons. For instance, 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. Repairing a machine takes time. Meanwhile the rest of the production facility continues to process parts. 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. 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 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.

10

From the above it is clear that apart from *repairing immediately* (RI) other policies should be considered when a machine breaks down. One option is to *stand-by* (SB). 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. Another option is to simply *switch-off* (SO) the machine and leave the repair until, say, the end of the shift.

Having decided what course of action to take, the group leader decides who must act. The engineers who can be asked to repair a machine are as follows: group leader (L1), second skilled engineer (L2), first semiskilled engineer (L3), second semiskilled engineer (L4) or unskilled engineer (L5). On occasions the group leader finds it necessary to seek advice from the production manager (*ask production manager* – APM).

Apart from determining what to do and who should do it, there is a third level of decision: whether to plan a repair. Sometimes repairs provide temporary fixes and more thorough maintenance is required to fully rectify the problem. Also, when a similar fault keeps recurring, a thorough investigation may be required. Often this type of work is carried out over the weekend as part of the preventative maintenance work. The group leader can flag the need for more thorough maintenance by specifying a *planned repair* (PR) and *when the planned repair* (WPR) should take place.

Given the above, the row vector $\mathbf{D}_{i,j}$ in this particular decision-making process includes the following elements:

$$\mathbf{D}_{i,j} = [\text{RI } \text{SB } \text{SO } \text{L1 } \text{L2 } \text{L3 } \text{L4 } \text{L5 } \text{APM } \text{PR } \text{WPR}]$$

With the exception of WPR, each element of the above vector 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 is not being asked to act. On the contrary, 1 means that the decision-maker is taking that 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 $\mathbf{D}_{1,1}$ would be as follows:

$$\mathbf{D}_{1,1} = [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

The decision variables were identified through discussions with the plant manager and the maintenance team, and through observation of working practices. The list of variables was refined by obtaining feedback from sessions in which the group leaders (one from each shift) interacted with the simulation model (stage 2 of the methodology). After three iterations of simulation model interaction, followed by enhancements to the model and the interface, an agreed list of decision variables as explained above was reached.

A similar procedure was used to identify the decision attributes. These attributes were as follows:

- Type of fault
- Estimated repair time
- Machine number
- Time of day
- Number of engines produced so far this shift
- Engines waiting on the conveyor before the machine
- Number of heads in the buffer
- Number of breakdowns on this machine today
- Number of breakdowns on this machine this month
- Number of breakdowns of this type on this machine today
- Number of breakdowns of this type on this machine this month

The number of heads in the buffer attribute refers to the feed of cylinder heads from a sub-assembly line that joins the main engine assembly line. The group leaders appear to take account of the number of cylinder heads queuing for assembly to an engine in making maintenance decisions.

The attributes listed above are the elements of the attribute vector $\mathbf{A}_i$. So, for example, assuming that at the first decision point the system has the following attribute levels:

- The fault type code is 300
- The estimated repair time is 5 minutes
- Machine 1130 has broken down,
- The time is 13:23
- The number of engines produced this shift is 549
- There are 3 engines on the conveyor before the machine
- There are 9 heads in the buffer
- There have been no breakdowns on this machine today
- There have been 2 breakdowns on this machine this month
- There have been no breakdowns of this type on this machine today
- There have been no breakdowns of this type on this machine this month

the attribute vector $\mathbf{A}_1$ would be as follows:

$$\mathbf{A}_1 = [300 \ 5 \ 1130 \ 13.23 \ 549 \ 3 \ 9 \ 0 \ 2 \ 0 \ 0]$$

Although it is possible to identify the types of decisions the group leaders might make via observation and discussion, it is another matter 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 the group leaders showed they are unable to directly express this knowledge. This is a well-known problem in knowledge management, summed up in Polanyi's comment "we know more than we can tell" [Polanyi, 1966]. Any method for knowledge elicitation has to address this issue [Kidd, 1987]. Here this issue is tackled in stage 2 of the methodology.

Stage 2: Data Collection

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 stopped when a decision point was reached, that is, when a machine breaks down. A Visual Basic front end was invoked and informed the user about the state of the system by reporting all the relevant attributes. The front end is shown in figure 2. The left hand window provides information on the decision attributes, the right hand window 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.

Figure 2 *Visual Basic Front End for the Ford Engine Assembly Model*
XXX spelling error - 'waiting' - Thanos
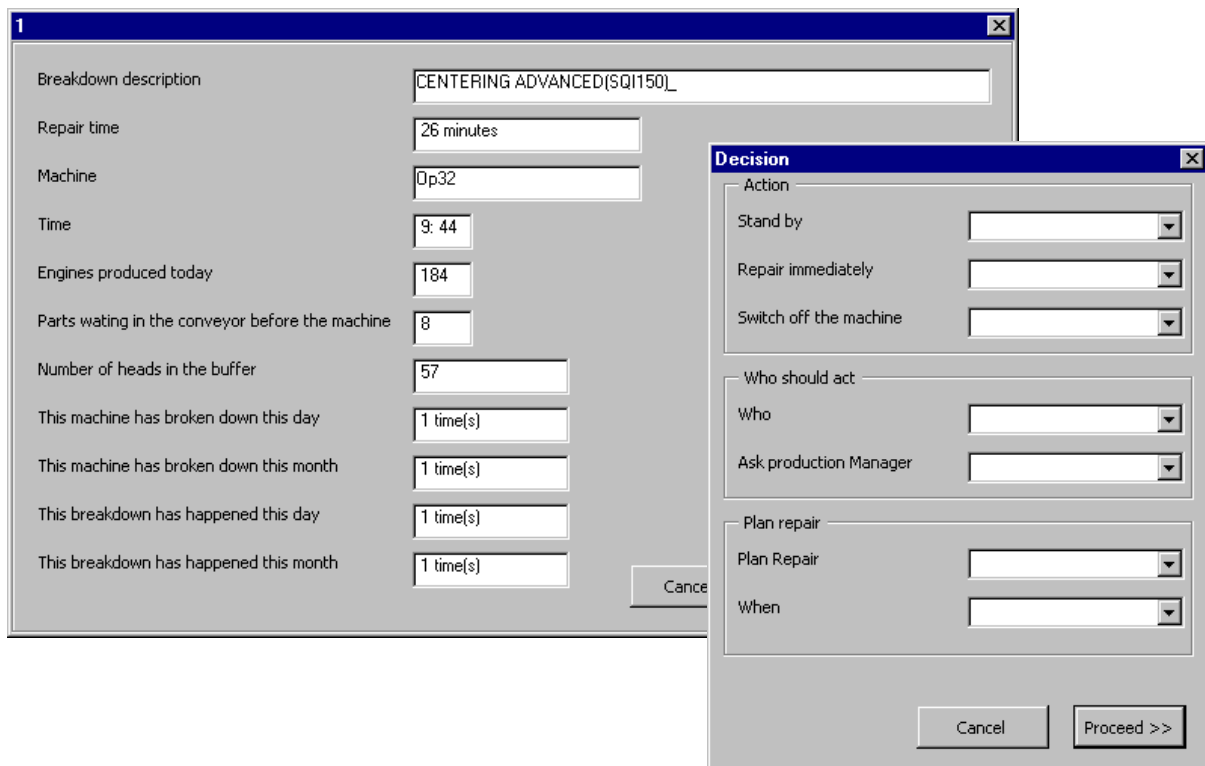
The user enters his/her decision and clicks on the 'proceed' button. The simulation then continues with the decision being implemented within the model. At each decision point (i) the simulation records the decision attributes ($\mathbf{A}_i$) and the decision variables ($\mathbf{D}_{i,j}$) in a data file.

Because the engine plant has automatic monitoring systems, it was possible to obtain a trace of historic data on machine failures and repairs over a three-month period. Initially this trace of data was used to drive the simulation model. It was found, however, that the decision-maker soon became bored when interacting with the simulation since the trace contained many repetitive decision scenarios. As a result the trace was adjusted by removing repetitive examples. This meant that the decision-maker was presented with fewer more interesting and extreme examples.

The slow run-speed of the simulation also led to decision-maker fatigue as he would have to wait for some time between each decision point. Further to this, it had been observed that the decision-makers made no reference to the simulation display, but simply to the Visual Basic front end. From this it was assumed that the front end contained all the information they required. As a result, the simulation was used to generate a series of decision scenarios up-front, which were then presented to the decision-makers through the front end without the simulation. This had the benefit of speeding the process of data collection and reducing decision-maker fatigue.

Knowledge elicitation sessions were carried out with three decision-makers (DM1, DM2, DM3); the group leaders for this section of the line from each shift. Each decision-maker was presented with 63 scenarios.

Stage 3: Determining the Decision-Makers' Decision-Making Strategies

Four AI approaches were applied to the data collected from stage 2. These were: artificial neural networks [Bigus, 1996], logistic regression [Pampel, 1996] and two rule induction methods (ID3 [Mingers 1986, 1987] and CART [Breiman et al., 1984]). The analysis focused only on the first decision, what to do (RI, SB, SO). Table 1 summarises the results of the analysis, showing the number of decisions that were not correctly classified using each AI method. The software in which the AI method was implemented is also noted.

Table 1 *Number of Example Decisions Correctly Classified by the Four AI Methods*

| AI Method | Software used | Decision-maker | | |
		DM1	DM2	DM3

Artificial neural network	Matlab	44/63	53/63	51/63
Logistic regression	SPSS	63/63	63/63	63/63
ID3	XpertRule	63/63	63/63	63/63
CART	SPlus	58/63	59/63	60/63

The artificial neural network performed particularly badly, misclassifying up to one third of the decisions. This is not unexpected since it is known that neural networks perform poorly when there are a limited number of examples [XXX Bigus, 1996]. A number of combinations of layers and nodes were tried, eventually settling upon a three layer network. Although further experimentation with the network structure may have given a better fit (fewer misclassifications), this was not carried out since the network was performing so poorly.

Logistic regression was used instead of standard multiple regression analysis because the decision variables are binary. This regression gave a perfect fit. After investigating alternative combinations of independent variables, the best fit was obtained with just five decision attributes: estimated repair time, machine number, number of engines produced so far this shift, engines waiting on the conveyor before the machine and number of heads in the buffer.

Of the two rule induction methods the ID3 algorithm achieved the best result in terms of correct classification. After trying different combinations of decision variables it was found that, depending on the decision-maker, two or three variables were required to be able to classify all the example decisions. The decision variables involved were: estimated repair time, machine number, and number of heads in the buffer.

As a result of this investigation the three decision trees (one for each decision-maker) obtained from the ID3 algorithm were selected for implementation in stage 4. These were selected on the basis of their success in classifying the example decisions, their ease of interpretation and their explanatory power. It was felt that although the logistic regression gave the same success in classifying the examples it did not perform so well on the later two elements.

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

The simulation model in Witness [Lanner Group, 2003] was linked to the expert systems software XpertRule [Attar Software, 2000] as described by Robinson et al [2003]. When a machine failure occurs in the simulation model, the simulation stops and the expert system is invoked. The decision is returned to the simulation model and the run continued implementing the decision that has been taken.

The simulation was run for a period of 112 days, giving 111 days of data after a warm-up period of one day. The daily throughput results (assuming only one shift per day) for the three decision-makers and the original logic (base) used in the model, always repair immediately, are shown in figure 3. The data have been scaled by a multiplier for reasons of confidentiality.

Figure 3 *Results of Simulation Linked to Expert System Representation of Decision-Makers*
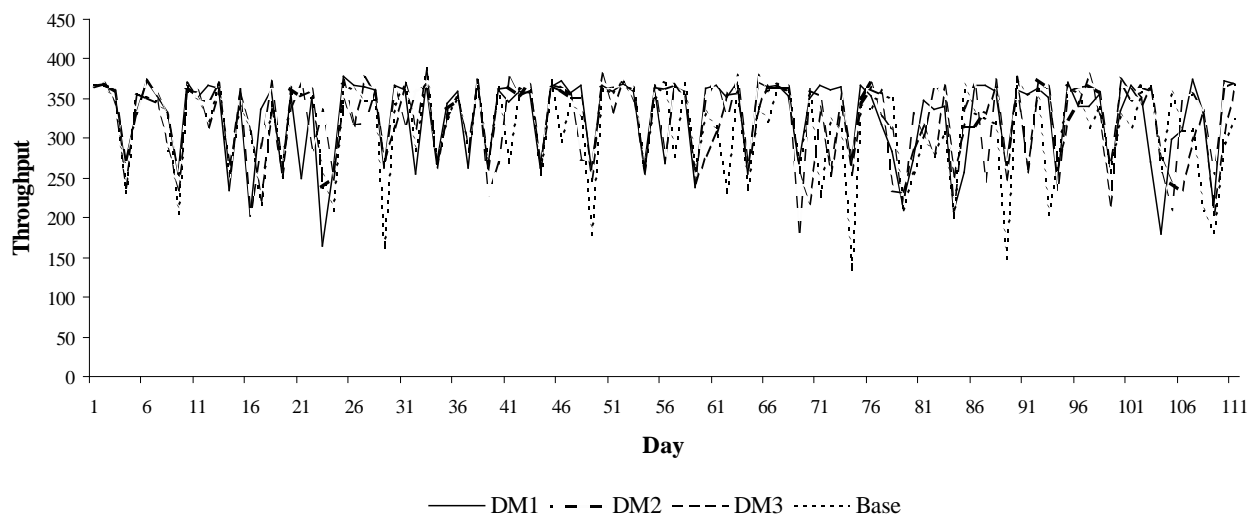
Albeit difficult to interpret the time-series in figure 3, the key point to note is that the series are different for each decision-maker, showing that the decision-making strategies have an impact on plant performance. It is notable that there are a number of days on which the strategy employed in the base model (repair immediately) performs more poorly than any of the decision-makers. This is generally due to a long breakdown for which a stand-by decision would be more effective in maintaining throughput.

17

Two statistical tests are performed on the data in figure 3. The first is a comparison of the mean differences in daily throughput between each decision-maker and between the decision-makers and the base model. The second is a regression analysis aimed at determining whether the time-series are correlated or whether there are differences in the throughputs achieved on a daily basis.

Table 2(a) shows the mean and standard deviation of the daily throughput for the four simulation runs. This suggests that DM1 and DM2 might perform slightly better than DM3 and the base model. It is notable that the standard deviation in throughput is much lower for DM2, which is likely to be preferred.

Table 2(b) shows paired-t confidence intervals for the differences between the results for the three decision-makers. According to the Bonferroni inequality a significance level of 1.67% (5%/3) should be used to give an overall confidence level of 95%, because three confidence intervals are being calculated [Law and Kelton, 2000]. Further to this, the data in the time-series have been batched in order to assure independence. Using Fishman's procedure [Fishman, 1978], a batch size of two is sufficient to assure a reasonable level of independence. The results show that the differences in the results for the three decision-makers are not significant.

Meanwhile, the results in table 2(c) show the paired-t confidence intervals for the differences between the results for each decision-maker and the base model. These are calculated using the same procedure as above. What this shows is a significant difference between both DM1 and DM2 and the base model. In both cases, the decision-makers outperform the decision-making strategy in the base model. This suggests that in a simulation model of the engine assembly plant that aims to assess plant performance in the face of machine failures, an appropriate representation of maintenance strategies is important.

Table 2 *Comparison of Mean Performance for each Decision-Making Strategy*

(a) Mean Daily Throughput and Standard Deviation

	DM1	DM2	DM3	Base
Mean daily throughput	325.53	325.15	318.87	312.06
Standard deviation	36.28	28.54	34.34	39.58

(b) 98.33% Paired-*t* Confidence Intervals for Differences in Daily Throughput between each Decision-Making Strategy

	DM2	DM3
DM1	-8.43, 9.19 (no sig. difference)	-3.80, 17.11 (no sig. difference)
DM2		-2.11, 14.66 (no sig. difference)

(c) 98.33% Paired-*t* Confidence Intervals for Differences in Daily Throughput between each Decision-Making Strategy and the Base Model

	Base
DM1	0.84, 26.10 (DM1>Base)
DM2	1.21, 24.98 (DM2>Base)
DM3	-4.04, 17.66 (no sig. difference)

Table 3 shows the results of regression analyses comparing the daily throughput achieved by each decision-maker. If there is a direct correspondence between two decision-makers, then it would be expected that the R-squared and beta (slope) values would both be 1. The results show the R-squared value for each comparison and confidence intervals for the beta values. Again, significance levels of 1.67% have been used, giving an overall level of confidence of 95%. Not only is there merely a moderate correlation in the data with all R-squared values being less than 0.5, but also the beta values suggest that there is not a direct correspondence between the throughput achieved by each decision-maker on a daily basis.

Table 3 *Comparison of Throughput Performance on a Daily Basis: Regression Analysis*

	DM2		DM3	
DM1	R-squared	0.49	R-squared	0.47
	98.33% CI	0.52, 0.83	98.33% CI	0.49, 0.81
DM2			R-squared	0.47
			98.33% CI	0.51, 0.85

The extent to which these results show differences between the decision-makers may be limited because the decision-making on only one section of the line has been represented. If the decision-makers on the other three sections were also represented, the differences may be much more marked.

Stage 5: Seeking Improvements

No formal methods were used in this case example for seeking improvements in the decision-making strategies. There may be some improvement if all group leaders adopted the strategies of DM1 or DM2. Any thorough search for improvements would have to look at decision-making across all sections of the assembly line. Without knowledge of the strategies that are employed on the rest of the assembly line, this was not possible.

One unexpected way in which the methodology helped to improve decision-making came about during data collection (stage 2). During a data collection exercise one of the group leaders who was relatively new to the role sought advice from other members of the team. In this way he was able to learn alternative approaches for dealing with specific situations. As such, the interactive simulation acted as a training tool.

Further work needs to investigate methods for seeking improvements. This should include informal approaches, such as comparing strategies and making incremental changes. It could also include a more formal search using, say, heuristic methods, or the use of rule induction to choose between different strategies.

Concluding Discussion

In applying the KBI methodology a number of strengths and weaknesses have been identified. The use of VIS for knowledge elicitation has a number of benefits. First, it is less time consuming than observation of the real system 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 decision situations. Finally, because the modeller has control over the decision scenarios that are presented to the decision-maker, it is possible to guarantee that decisions are recorded for a wide range of scenarios. Should important scenarios be overlooked during data collection, it is always possible to perform further knowledge elicitation sessions in which those scenarios are presented.

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. In some circumstances the simulation may not be capable of reporting all of the decision attributes. In the Ford case the group leaders took the physical condition of the machine into account when determining what course of action to take. This attribute is all but impossible to simulate in a model of this nature. Fortunately it was possible to use the estimated repair time as a meta-attribute that would in practice include knowledge of the machine's condition.

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.

Beyond the use of VIS for knowledge elicitation (stage 2) there are many other issues that need further investigation. Could a more formal process be employed for understanding the decision-making process (stage 1)? Perhaps soft OR methods might be applicable at this stage. Which AI approaches are most effective for learning, representing and improving decision-making strategies (stage 3)? Perhaps different methods should be used for different purposes. The validation of the AI representation also needs to be investigated. When running the simulation linked to AI, what should happen if a scenario occurs for which there is no decision logic (stage 4)? How would such an occurrence be identified? Could some form of interpolation be used to fill holes in the decision-logic (for a decision tree, this would require the identification of a further supporting model)? Finally, there is the need to investigate stage 5 of the methodology in greater detail than has so far been possible.

A second research project is under way which is investigating in more detail the knowledge elicitation aspects of the methodology. Again based at a Ford engine assembly plant, the methodology is being applied to a hot test area. The objective is to compare alternative mechanisms for eliciting knowledge from decision-makers using VIS in terms of their efficiency and effectiveness. Specific issues to be addressed are the level of visual display, the nature of the interactive interface and the methods for generating decision scenarios.

Acknowledgements

References

XXX ref numbering – new refs [], more refs to add – search on [] and (). Format to JORS standard.

Baines, T.S. and Kay, J.M. (2002). Human Performance Modelling as an Aid in the Process of Manufacturing System Design: A Pilot Study. *International Journal of Production Research*, **40** (No. 10): 2321-2334.

Hurrion, R.D. (2000). A Sequential Method for the Development of Visual Interactive Meta-Simulation Models using Neural Networks. *Journal of the Operational Research Society*, 51 (6), pp/ 712-719.

Pierreval, H. (1992). Expert Systems for Selecting Priority Rules in Flexible Manufacturing Systems. *Expert Systems with Applications*, 5, pp. 51-57.

Moffat, J. (2000). Representing the Command and Control Process in Simulation Models of Conflict. *Journal of the Operational Research Society*, 51 (4), pp. 431-439.

Kidd AL, ed. *Knowledge Acquisition for Expert Systems: A Practical Handbook*. 1987, Plenum: New York.

Polanyi M (1966). *The tacit dimension*. Doubleday: Garden City, N.Y.

Bigus JP (1996). *Data mining with neural networks : solving business problems--from application development to decision support*. McGraw-Hill: New York ; London.

Pampel FC (2000). *Logistic Regression : A Primer*. Sage Publications: London.

Mingers J (1986). Expert systems - experiments with rule induction. *Journal of the Operational Research Society* **37** (11)**:** 1031-1086.

Mingers J (1987). Expert systems - rule induction with statistical-data. *Journal of the Operational Research Society* **38** (1)**:** 39-47.

Breiman L, Friedman JH, Olshen RA, and Stone CJ (1984). *Classification and Regression Trees*. Wadsworth: Belmont, CA.

Attar Software (2000). *XpertRule KBS Reference Manual.* Attar Software, Leigh, UK.

Robinson, S., Edwards, J.S. and Yongfa, W. (2003) Linking the Witness Simulation Software to an Expert System to Represent a Decision-Making Process. *Journal of Computing and Information Technology*. XXX Forthcoming

[1] Doukidis GI and Angelides MC (1994). A framework for integrating artificial intelligence and simulation. *Artificial Intelligence Review* **8** (No. 1): 55-85.

[2] Doukidis G and Paul R (1985). Research into expert systems to aid simulation model formulation. *J Opl Res Soc* **36** (No. 4): 319-325.

[3] Hurrion RD (1993). Representing and learning distributions with the aid of a neural network. *J Opl Res Soc* **44** (No. 10): 1013-1024.

[4] Balmer D and Paul RJ (1986). CASM - the right environment for simulation. *J Opl Res Soc* **37** (No. 5): 443-452.

[5] Mathewson S (1989). Simulation support environments. In: Pidd M (ed). *Computer Modelling for Discrete Simulation.* Wiley: Chichester, pp 57-100.

[6] Doukidis GI (1987). An anthology on the homology of simulation with artificial intelligence. *J Opl Res Soc* **38** (No. 8): 701-712.

[7] Deslanders V and Pierreval H (1991). An expert system prototype assisting the statistical validation of simulation models. *Simulation* **56** (No. 2): 79-89.

[8] Hurrion RD (1991). Intelligent visual interactive modeling. *European Journal of Operational Research* **54** (No. 3): 349-356.

[9] Hurrion RD (1992). Using a neural network to enhance the decision making quality of a visual interactive simulation model. *J Opl Res Soc* **43** (No. 4): 333-342.

[10] Hurrion RD (1993). Using 3D animation techniques to help with the experimental design and analysis phase of a visual interactive simulation project. *J Opl Res Soc* **44** (No. 7): 693-700.

[11] Hurrion RD (1997). An example of simulation optimisation using a neural network metamodel: finding the optimum number of kanbans in a manufacturing system. *J Opl Res Soc* **48** (No. 11): 1105-1112.

[12] Flitman AM and Hurrion RD (1987). Linking discrete-event simulation models with expert systems. *J Opl Res Soc* **38** (No. 8): 723-734.

[13] O'Keefe RM (1989). The role of artificial intelligence in discrete-event simulation. In: . Widman LE, Loparo KA and Neilsen NR (eds). *Artificial Intelligence, Simulation and Modeling*. Wiley: New York, pp 359-379.

[14] Williams T (1996). Simulating the man-in-the-loop. *OR Insight* **9** (No. 4): 17-21.

[15] Lyu J and Gunasekaran A (1997). An intelligent simulation model to evaluate scheduling strategies in a steel company. *International Journal of Systems Science* **28** (No. 6): 611-616.