KBS Methodology as a framework for Co-operative Working

John Kingston

AIAI-TR-130

August 1993

This paper was presented at "Expert Systems 92", the annual conference of the BCS Specialist Group on Expert Systems, held on 16-17 December 1992 at Churchill College, Cambridge. It was published in the proceedings of that conference: "Research and Development in Expert Systems IX", Cambridge University Press, 1992.

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

Abstract

This paper describes the development of the Injection Moulding Process Expert System (IMPRESS). The IMPRESS system diagnoses faults in injection moulding machinery which lead to dirt or other contamination appearing in the plastic mouldings which are produced. This KBS has recently been put into use at Plastic Engineers (Scotland) Ltd, and is proving useful both as an expert assistant when technical help is otherwise unavailable, and as a training aid.

The IMPRESS system was built by a member of Plastic Engineers' staff with assistance from a KBS consultant. It was decided that the project would be based around a KBS methodology; a 'pragmatic' version of the KADS methodology was chosen. The methodology was used not only to formalise and guide the development of the KBS itself, but also to act as a framework for dividing the work between the two members of the project team. By gaining an understanding of the methodology, the staff member from Plastic Engineers was able to understand the knowledge analysis and KBS design documents produced by the consultant, and to use these documents to implement part of the KBS, both during the development of the system and when system maintenance was required.

The use of a methodology for this project on this project had both benefits and weaknesses, which are discussed at the end of the paper.

1 Introduction

In January 1992, Plastic Engineers (Scotland) Ltd obtained funding from Scottish Enterprise to help them in the development of a knowledge based system (KBS) for fault diagnosis. Plastic Engineers manufacture precision plastic mouldings, such as casings for PCs, or control panels for video recorders. They have a reputation for high quality, which they want to maintain. However, from time to time, problems with their injection moulding machines mean that substandard mouldings are produced, and these have to be scrapped to maintain the reputation for quality. While Plastic Engineers have technicians who are very competent at solving these problems, these technicians have a variety of roles to perform. If a technician is working on an urgent task, or is absent through holidays or illness, it may take some hours before diagnostic expertise is available. Shift leaders are able to provide some backup to technicians, but they have even more demands on their time than the technicians do. As a result, there are times when no-one with diagnostic knowledge is available, particularly during some night shifts.

After attending a seminar organised by AIAI and the Scottish Office in the summer of 1991, the idea of building a KBS to help with the diagnostic process was born. The project was set up in January 1992 with Plastic Engineers releasing one member of staff to work on the project for two days per week. This member of staff [JM] was a newly recruited graduate in Polymer Technology with knowledge of the process of injection moulding, but very little computing experience. AIAI were engaged to provide JM with initial training in KBS programming, knowledge elicitation and knowledge engineering (a total of 7 days' training) and then to provide 15 man days' consultancy spread over the 4-month duration of the project. The intention was that by the end of the project, JM would be fully conversant with the techniques used to develop the KBS, and would therefore be able to maintain the system if any changes were needed after installation.

AIAI decided to use a methodological approach to this project. The use of KBS methodology in the commercial world is still in its infancy, but AIAI were sufficiently convinced of the benefits of methods to use a simplified version of the KADS methodology on this project. However, in this project, the methods were used not only to formalise and guide the development of the KBS itself, but also to act as a framework for the division of labour and transfer of KBS expertise. This paper describes the benefits and drawbacks of using a methodology in this way.

Before any development could take place, however, a number of factors needed to be established to ensure that the KBS project stood a good chance of success. These included:

- Economic considerations. Plastic Engineers do have a genuine problem with quality control - they scrap around 2% of their production each month. The KBS is likely to make a significant improvement to the availability of diagnostic expertise, and to the early detection of faults, thus reducing scrap rates.

- Technical considerations. Diagnosis is known to be a task type which KBS are well suited for; also, the technicians currently take between several minutes and a few hours to solve problems, so there are unlikely to be any stringent requirements for real-time problem solving.

- Personnel considerations. The project was initiated by Plastic Engineers' General Manager, so management support was assured. The users - the machine operators - are likely to appreciate any help their shift leaders can give them in diagnosing faults. However, the commitment of the shift leaders and technicians themselves was unclear, so the AIAI consultant [JK] made a presentation to these people, which included a demonstration of a very simple KBS which diagnosed three different faults in the plastic moulding process. While the underlying structure of this demonstration system was very shallow in its reasoning, and drew knowledge from just one day of knowledge acquisition, it was sufficient to convey the concept of a KBS to the shift leaders and technicians, and to excite their curiosity so that they began to ask questions about the capabilities of the system. This was deemed to be sufficient commitment for the project to proceed.

The project was named IMPRESS (the Injection Moulding PRocess Expert SyStem project).

2 The framework of the IMPRESS project

The KADS methodology divides the process of KBS development into three phases: knowledge elicitation and analysis, KBS design and KBS implementation. The IMPRESS project was set up with a number of intermediate milestones accompanied by deliverables; these milestones were based around the phases specified by KADS. The phases specified in the project plan were:

- Knowledge elicitation and analysis - 6 weeks.
- KBS design - 4.5 weeks
- KBS implementation - 4.5 weeks
- Testing and installation - 2 weeks

The workload was divided between JM and JK in a manner which was intended to get the project completed within the deadline, but also to give JM a sufficient awareness of KBS development and the contents of the IMPRESS system to enable him to update it. The policy pursued was for both JK and JM to attend knowledge elicitation sessions; then for JK to perform the knowledge analysis and KBS design while JM undertook background reading on KADS so that he understood the deliverables which JK produced; and finally for JM to undertake the lion's share of the implementation, and to carry out user acceptance testing, any consequent alterations, and installation. The plan was adhered to fairly closely, and JM was indeed able to make alterations to the KBS himself in response to comments from the users.

3 Progress of the project

3.1 Knowledge Elicitation

Knowledge elicitation for the IMPRESS system was carried out at Plastic Engineers' premises in Ayrshire. The first interview was with one of the shift leaders, who was asked to provide a general overview of the problems which arise in the plastic moulding process. The interview was guided using the "laddered grid" knowledge elicitation technique [6]. This technique supplies a number of template questions which are designed to prompt experts to supply further information about a taxonomic hierarchy - for example, the question "Can you give me some examples of *Class*" will supply information about instances or subclasses of the class *Class*. The technique can also be used to elicit procedural information. In the interview with the shift leader, the resulting grid comprised both a detailed description of some of the faults which arise in the plastic moulding process, including descriptions of different symptoms and associated faults, and also explanations and corrective action for some faults. While it is not desirable for analysis purposes for the expert to be allowed to mix taxonomic and procedural information in his replies, this interview nevertheless provided a concise introduction to the domain and the diagnostic task.

The next interview was with the Quality Manager, who provided a breakdown of the five main categories of fault. These categories are

- Contamination - dirty marks of some kind on the final moulding
- Shorts - certain parts of the mould do not fill with plastic
- Burns - discolouration due to plastic being overheated
- Degate - human error when trimming with a knife
- Others

The Quality Manager keeps detailed statistics of the number of times each fault has occurred, and how long it takes to solve. From examination of these statistics, it became obvious that contamination was the most frequently occurring problem, and that contamination problems took an average of almost 2.5 hours to solve. Based on this information, it was decided that the KBS would initially be limited to diagnosing contamination problems only.

All other knowledge elicitation interviews were conducted with technicians, who are the day to day diagnostic experts. Most of these interviews used a "20 questions" knowledge elicitation technique [1]. This technique is normally used after several knowledge elicitation sessions, because it requires the knowledge engineer to be fairly familiar with the task. The knowledge engineer selects a potential fault, which the expert is required to diagnose; the expert does this by asking questions, which the knowledge engineer answers. As JM had some knowledge of the injection moulding process and of Plastic Engineers' machinery, it was possible to use this technique from a very early stage.

A typical "20 Questions" session is shown below. The hypothesised fault was dust entering the machine via the drier which dries the raw material. The technician was told that there were "black specks on the moulding". JM's answers to the technician's questions are shown in brackets.

```
What's the tool? [155]
Where are the marks? [Back face, sides - all over]
```

```
How long has the job been running? [2 days]
Has the problem been present since start up? [Yes]
Is the problem getting worse? [Yes]
Have you cleaned the shims? [Yes, it caused a little improvement, but
the problem recurred]
Is the temperature unstable, or too high? [No]
Check the thermocouplings [OK]
Check the condition of the screw, and look for black specks on the screw [OK]
```

On being told the answer, the technician commented that dust from the drier was almost never a problem because of the reliability of the drier's filtration system.

The technician was then asked to explain his reasons for asking each question. The information which was extracted from the conversation described above and the subsequent explanation included:

- Possible faults include dirty shims, incorrect temperature settings, loose thermocouplings, and dirt on the screw.
- Some faults are more prevalent on certain machine tools - usually tools which produce large mouldings.
- If the marks had appeared only on the bottom edges of the moulding, this would have been a very strong indicator of one particular fault.
- Certain faults only occur shortly after the machine has been started up. Many of these are due to the machine not being cleaned properly before being shut down.
- If the problem only occurs for a short time, then the fault is likely to be contamination in a single batch of raw material.
- If the problem is getting worse, then it is likely to be due to some material which is trapped in the machine and slowly degrading
- Dust in the drier hardly ever causes a problem because it is filtered out

The "20 Questions" technique proved to be very helpful for eliciting diagnostic information, with a lot of useful information obtained in a concise format in a short period of time.

3.2 Knowledge Analysis

The technicians' knowledge divides into three main categories:

- Declarative knowledge - the workings of the machine, and knowledge of all faults which may occur.

- Procedural knowledge - knowing how to test for and how to fix faults.
- Control knowledge - performing tests in a sensible order.

The declarative and procedural knowledge was relatively straightforward to extract from the results of the "20 Questions" sessions, but the control knowledge required a little more thought. It was eventually determined that the likelihood of a fault occurring, and the time required to perform a particular test, were the most important factors in deciding the order in which tests should be performed. For example, in the "20 Questions" session quoted above, the technician asked about the condition of the screw last, because it takes a couple of hours to dismantle the machine sufficiently to expose the screw, and he did not ask about dust in the drier at all, because it is such a rare fault.

It turned out that there are quite a number of rare faults. However, as JM spent much of his time on the shop floor when he was not working on the KBS, it was decided that JK would press ahead with the analysis phase while JM completed the elicitation of all possible faults from the experts. The final KBS contains about 40 faults (broken down into five subclasses) and a similar number of tests.

3.3 KBS design, implementation, testing and installation

The analysed knowledge was transformed into a KBS design using techniques based on the KADS methodology (these techniques are outlined in section 4). The KBS was then implemented in KAPPA-PC version 1.2 on an Apricot 486 PC. The resulting design suggested that faults, tests, and test results should be represented using individual objects, while inference should be implemented primarily using a mixture of rules and functions, with a little use of object-oriented methods and demons. However, it transpired that some of the desired rule functionality was unavailable in KAPPA-PC; it also became clear that the time taken to execute a rule which matched on a set of objects was similar to the time taken for a function to iterate over the same objects. As a result, it was decided that rules would not be used at all, and so much of the inference in the IMPRESS system was implemented using functions.

The KBS was subjected to testing by developers concurrently with the implementation of the user interface, and was installed in the first week of August 1992. At the time of writing, few firm results were available, because there have been relatively few occasions since the installation of the KBS when there has been no technical expert available to answer questions. However, the fact that the system can be used "off-line" has been appreciated, and the KBS has been used several times for training purposes by interested machine operators.

4 Using KADS for the IMPRESS project

The KADS methodology for KBS development [2] is intended both to guide and to formalise KBS development. To this end, it provides guidance on obtaining knowledge, analysing it, and transforming it into a detailed design for an implemented KBS. The IMPRESS project did not use the KADS methodology in its entirety, but instead followed the "pragmatic KADS" approach described in [3].

4.1 Knowledge analysis: interpretation models

Once some knowledge has been acquired, the KADS methodology recommends selection of an *interpretation model*. Interpretation models are task-specific breakdowns of the inferences and items of knowledge required in a typical task of that type. These models are intended both to formalise acquired knowledge and to guide further knowledge acquisition. For the IMPRESS system, it was obvious from the start that the task type was diagnosis; however, KADS offers several different interpretation models for different methods of performing diagnosis. Eventually, it was decided that the interpretation model for *systematic diagnosis* was the most appropriate. This model is shown in Figure 1 below; the ovals are known as "inference functions", and the boxes as "knowledge roles"[1].

[1]Strictly speaking, Figure 1 represents only one component of an interpretation model. However, under "pragmatic KADS", the other component is not used, and so the structure shown in this diagram is described as an interpretation model throughout this paper.

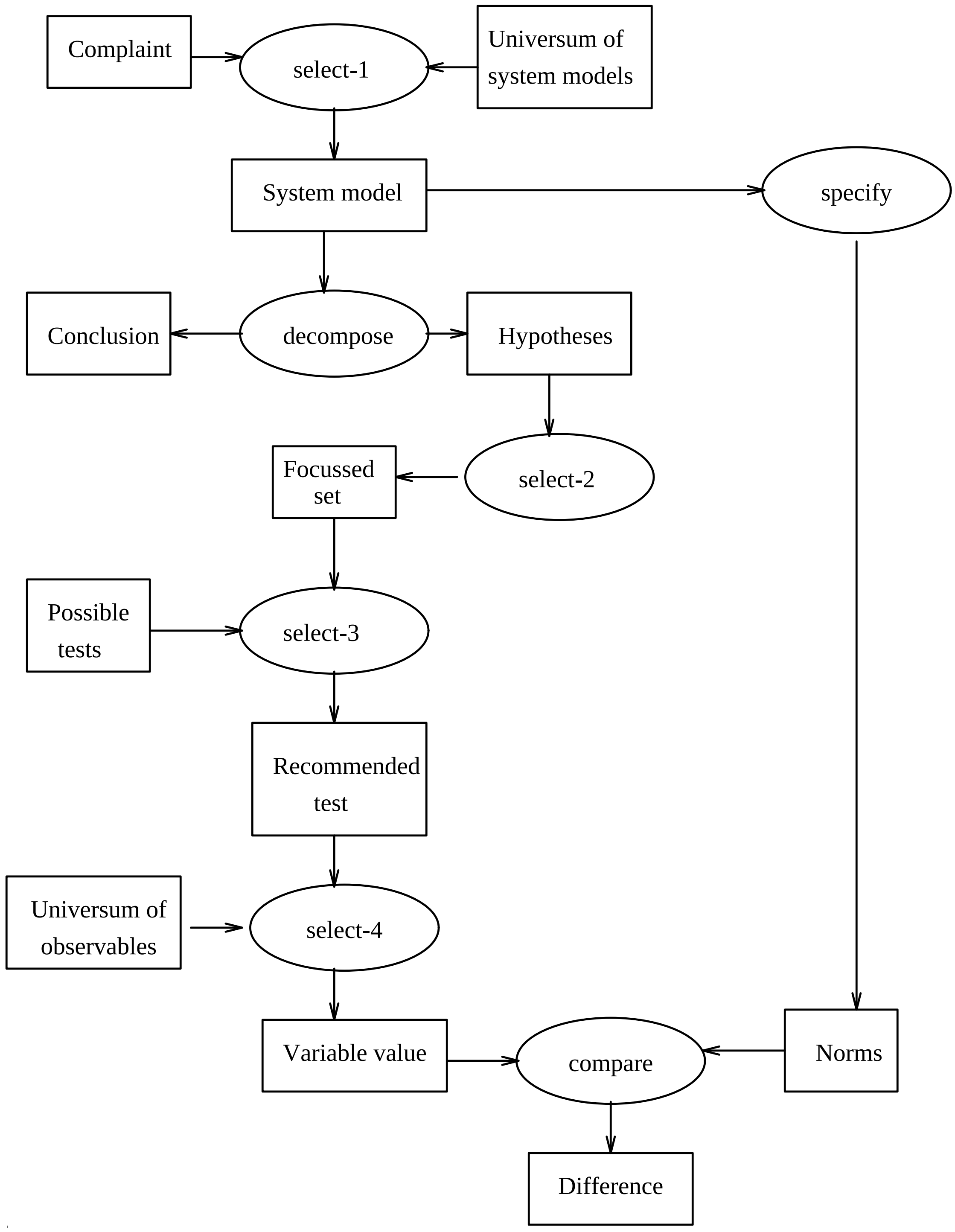

Figure 1: Interpretation model for systematic diagnosis

This model represents the inference which is expected to be performed when a task involving systematic diagnosis is executed. For example, if a user reports a problem with a machine, it is expected that a particular system model representing the correct operation of that machine will be selected, and a number of faults will be suggested. Based on a 'focussed' subset of these faults, a number of characteristics of the machine will be measured and compared with their expected values in the system model.

This model was then adapted to the domain of the IMPRESS system, as shown in Figures 2 and 3 below (Figure 3 is an expansion of the **select-1*** inference function in Figure 2), to produce a problem-specific *inference structure*. This inference structure indicates that the IMPRESS system will identify a set of possible faults (hypotheses) based on the reported contamination problem. A test is then recommended, based on the likelihood of the hypotheses, the time required to perform a test and the time required to alter the state of the machine so that the test can be performed. Once it has been decided which test will actually be performed, the test is carried out, and the actual result is compared against a set of expected results (see below) in order to update the set of hypotheses.

It can be seen that the adaptation from the interpretation model to the inference structure involved a number of changes. Most of these changes are relatively minor, such as the removal of the focussing of the set of hypotheses into a smaller set; it was felt that the set of hypotheses was sufficiently small that such a step was not necessary. However, one of the changes implies a fundamental change to the approach taken to reasoning. This change involved the interpretation model's suggestion of comparing values against a system model, which is a *model-based* approach to KBS construction. While a model-based approach would have worked adequately for the IMPRESS system, it was felt that explicitly representing injection moulding processes was not worth the effort, primarily because all Plastic Engineers' machines operate in the same manner, and so only one "system model" would be required. Instead, it was decided that for every known fault, the expected results of each test would be represented. For example, if the fault was "Contamination of raw material due to the box of material being left open", then a check on the material currently being fed into the machine should produce the result *Contamination present*, while a check on a fresh box of material should produce the result *Contamination absent*. These values were explicitly represented, and compared against the actual results of tests, as shown at the bottom of Figure 2.

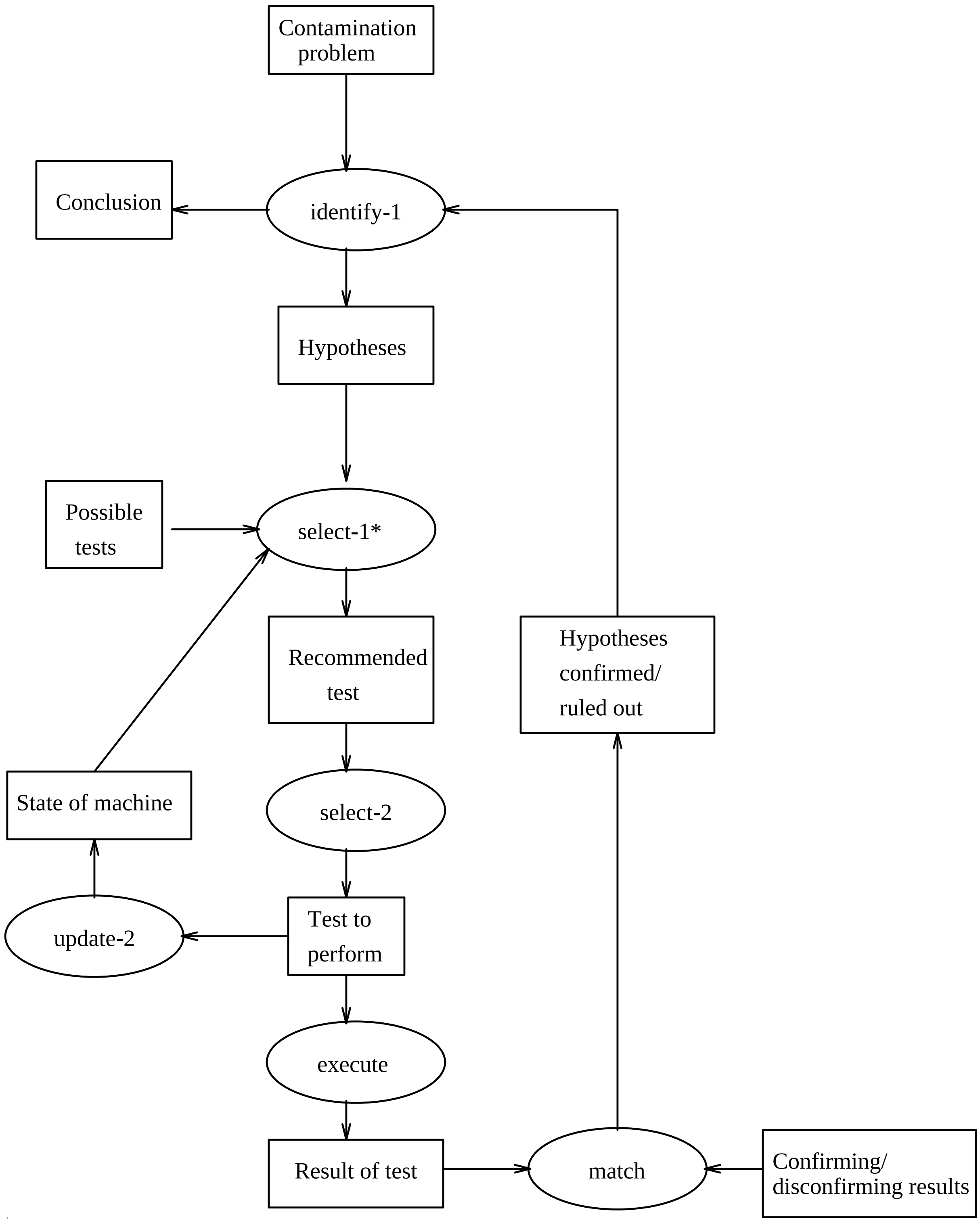

Figure 2: Inference structure for IMPRESS system

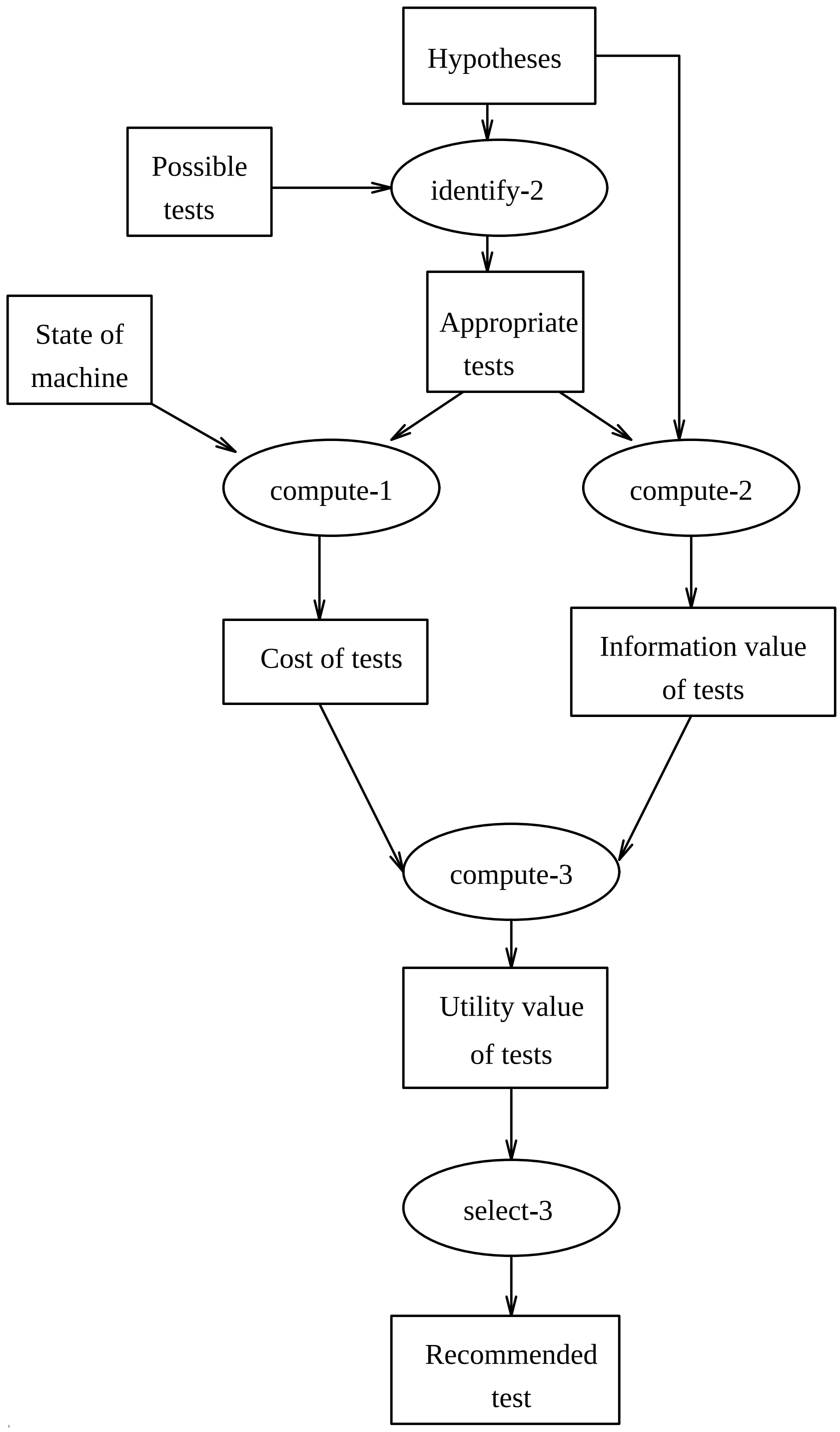

Figure 3: Inference structure for test selection in IMPRESS system

4.2 Further guidance provided by pragmatic KADS

The remaining stages of the pragmatic KADS analysis and design phases gradually extend and transform the knowledge which is represented in the inference structure into a detailed KBS design, with any design decisions being explicitly recorded. These stages are:

Knowledge analysis:

- The *task structure* identifies the flow of control between inference functions, and also identifies any inputs and outputs of the KBS.
- The *model of interaction*, an addition to the KADS methodology used by AIAI [4], assigns inference functions to the KBS, the user, or the two working together. The model of interaction is based on KADS' "model of cooperation", which is used to determine which overall task(s) should be performed by a knowledge based system. The model of interaction performs a similar function *within* a single KBS; it helps determine which of the inference functions should be performed by the system, which by the user, and which by the two working together. It also explicitly identifies every input and output within the system.

  The main decision made when developing the model of interaction for the IMPRESS system was that the selection of a test to perform would be done by the KBS and user in conjunction, rather than by the KBS alone; in other words, the KBS would recommend a test to perform, but the user would be free to reject the recommendation.

KBS design:

- *Functional decomposition* involves laying out the inference functions, knowledge roles and inputs/outputs in a single diagram, and identifying the data flow between them.
- *Behavioural design* involves the selection of AI "design methods", such as best-first search, blackboard reasoning, or truth maintenance, to implement each function in the functional decomposition. AIAI's pragmatic KADS approach makes use of a set of *probing questions*, based on the work of Kline & Dolins [5], to recommend design methods.
- *Physical design* involves the selection of rules, objects, or other low-level design techniques to implement the chosen design methods. This proved to be the most difficult of all the analysis and design stages, partly because the behavioural design stage did not produce many strong recommendations for particular design methods.

KADS recommends that the selection of a KBS implementation tool should be based on the results of this stage; however, an implementation tool has often been chosen by the time this stage of the project is reached, and so it is sensible if the capabilities of the KBS tool are borne in mind when performing physical design.

Once the physical design is complete, KADS suggests using conventional software engineering methods. While these methods are likely to work for implementation, they may not be adequate for verification and validation, which may differ significantly between a KBS and conventional computer programs [7].

4.3 Technology transfer using KADS

During the stages of knowledge analysis and KBS design, technology transfer was accomplished by introducing JM to KADS. This was achieved during JM's initial training. JM was also asked to read sections of the best current single reference on KADS[2]. With this background, JM was able to understand the deliverables from the analysis and design phases at a detailed level, and to use these deliverables as a basis for the implementation of the IMPRESS system.

The aim of using KADS for technology transfer was that JM would understand the KADS models sufficiently well that, should the occasion arise, he would be able to make a change to the inference structure and propagate the change through all the remaining stages in order to produce a revised physical design. This change would then be implemented in the KBS, and the revised set of models would serve as up to date documentation for the system. This purpose appears to have been achieved.

5 Benefits and weaknesses of methods for the IMPRESS project

The use of pragmatic KADS for the IMPRESS project provided a number of benefits, but also had some weaknesses. These are outlined below.

Benefits: The major advantage of KADS from the point of view of technology transfer is the large number of models which are produced during the development of the KBS. These models represent the KBS from a number of different viewpoints, so a novice stands a much greater chance of understanding the workings of the KBS from these models than from any single document describing the KBS. The variety of models also helps greatly when a new piece of knowledge or a new procedure must be added to the KBS, and it is difficult to decide where this new information fits into the previous structure. These models also force the KBS developer to document design decisions explicitly, which is almost essential for successful long-term maintenance, and can constitute a set of deliverables from each stage of the project for the management or project monitoring officer.

KADS itself has some particular advantages. The library of interpretation models is widely thought to be the most useful contribution of KADS to knowledge engineering, and it certainly provided a lot of assistance for the IMPRESS project. There is also some reasonably comprehensible background reading available on KADS which helps introduce novices to the methodology.

Weaknesses: Perhaps the biggest disadvantage of using KADS, when compared with a "rapid prototyping" approach to KBS development, is that implementation does not begin until relatively late in the project. While the preparation of a design which has been thought out and documented well provides plenty of justification for KADS' approach, late implementation carries disadvantages both for technical development and for technology transfer.

From the viewpoint of technical development, KADS' approach loses the advantages of iterative prototyping for knowledge acquisition and investigating possible implementation techniques. KADS does not rule out the use of prototyping as a knowledge acquisition technique, but it is time-consuming to build a prototype based on an uncertain system design which will eventually be thrown away, and it was decided that this approach was not worthwhile for a small-scale project such as the IMPRESS project. Iterative prototyping is also very useful for identifying omissions or misunderstandings in knowledge acquisition and analysis, and the fact that most of KADS' models are based on the analysed knowledge (directly or indirectly) means that errors in knowledge acquisition and analysis are costly, because they require almost all the models to be updated. A CASE tool for KADS would go a long way towards alleviating this difficulty.

From the viewpoint of technology transfer, KADS' approach means that a novice KBS programmer (JM in this project) is thrown into programming at the deep end, rather than being gradually introduced to implementation techniques as the prototype is built. While JM was given some training and programming exercises in KAPPA-PC while the analysis and design phases were being conducted, it is received wisdom that the only way to understand a KBS implementation tool fully is to use it to develop a full-scale KBS, and this project reinforced that belief. This unfamiliarity was a major contributor to the fact that the implementation phase overran by about 3 weeks, the only phase to show a significant deviation from the initial plan.

Two other features of KADS were noted which were minor disadvantages in the IMPRESS project:

- KADS provides little guidance on user interface design, which is something of a disadvantage since the development of user interfaces may take up a large proportion of the code and the development time for a KBS. For the sake of simplicity, the IMPRESS project used KAPPA-PC's built-in user interface facilities (menus, message boxes and text windows) to develop its user interface.

- The physical design stage should take into account the features of the chosen KBS implementation tool. KADS recommends that a tool should be chosen based on the results of the physical design stage, but in practice a tool has almost always been chosen before this stage. For example, the physical design for the IMPRESS system recommended the use of a series of demons on the slots of the **State of the machine** object to calculate the total time required for the machine to be put into a particular state. However, demons in KAPPA-PC do not return a value, so instead of using a return value, the technique had to be implemented using a global variable to accumulate the total time.

6 Conclusion

On the whole, the use of a methodology as a framework for technology transfer worked well on the IMPRESS project, and is recommended for other projects. However, a number of factors must be considered carefully when doing so:[2]

- Considerable effort is required to make sure that knowledge analysis is done properly, because of the effort required to correct errors at a later stage. In larger projects, or other projects where the knowledge to be acquired is particularly complex, it may well be worth developing a prototype to assist in knowledge acquisition.
- The implementation stage should be given at least as much time as the analysis stage, if not more, unless the chief programmer is **fully** conversant with the KBS implementation tool before the implementation stage is reached.
- Documentation should be prepared in a format which is fairly easy to update, since it is expected that the documentation will change over time.
- The features of the chosen implementation tool should be taken into account at the physical design stage (or equivalent stage in the chosen methodology).

[2]These comments assume that the methodology uses the three phases of analysis, design and implementation.

References

[1] Burton A M, Shadbolt N R, Rugg G and Hedgecock A P. Knowledge Elicitation Techniques in Classification Domains. In *Proceedings of ECAI-88: The 8th European Conference on Artificial Intelligence*, 1988.

[2] F. Hickman, J. Killin, L. Land *et al.* *Analysis for knowledge-based systems: A practical introduction to the KADS methodology.* Ellis Horwood, Chichester, 1989.

[3] J. Kingston. Pragmatic KADS: A methodological approach to a small KBS project. *Submitted to Expert Systems: The International Journal of Knowledge Engineering.* This paper is also available as AIAI Technical report AIAI-TR-110.

[4] J. Kingston. The model of interaction. *Newsletter of the BCS Methodologies Interest Group*, (1), August 1992. Also available from AIAI as AIAI-TR-115.

[5] Kline, P J & Dolins, S B. *Designing expert systems : a guide to selecting implementation techniques.* Wiley, 1989.

[6] Shadbolt N.& Burton, M. Knowledge elicitation. In J. Wilson and N. Corlett, editor, *Evaluation of Human Work: A Practical Ergonomics Methodology*, pages 321–346. Taylor and Francis, 1990.

[7] T.J. Lydiard. A survey of verification and validation techniques for KBS. *Knowledge Engineering Review*, 7(2), June 1992.