INTELLIGENT SYSTEMS SUPPORT FOR FIELD SERVICE ENGINEERS IN A FLEXIBLE MANUFACTURING ENVIRONMENT: THE STOVES PROJECT

Zhang, W., Coenen, F. P. and Leng, P. H.
Department of Computer Science,
The University of Liverpool,
Liverpool L69 7ZF
{wzhang,frans,phl}@csc.liv.ac.uk

Abstract

In this paper we present a short review of the aims and objectives of the "Stoves" project. This is a DTI sponsored *Foresight-Link* project which commenced in December 1999. The principal industrial partner is Stoves PLC (hence the project title). Additional industrial support is provided by Euro-Serv, VG-Elemental and NA-Software. In the first half of the paper we introduce the background to the project, and then goes on to discuss the nature of the proposed data structures required to support the intended end goals. In the second half of the paper we concentrates on the components which we expect to be presented once the data structure is fully "populated". In particular we discuss the intelligent systems components.

1 INTRODUCTION

It is becoming increasingly common for manufacturing industries to tailor their operation to be much more in line with their customer requirements than (say) 10 years ago. That is to say that instead of offering a range of (say) 6 versions of a product, starting with a "deluxe" model at the top and working down to an "economy" model at the bottom; individual customisation is offered for the entire product range. This serves to increase the desirability of the product and (it is hoped) will encourage a corresponding increase in sales. There are, however, a number of difficulties associated with this approach. One is that, although (say) 6000 versions of a product are on offer, the storage of data concerning each version becomes problematic. More seriously the quantity of maintenance data required to provide an appropriate level of after sales services becomes increasingly difficult to (a) provide (in an effective manner) and (b) update. This is exasperated when, as is often the case, the manufacturer is operating in a global market place. Thus although after sales services (i.e. maintenance) are seen as an important issue with respect to customer loyalty, there are issues concerning the provision of appropriate maintenance data that need to be addressed.

A similar problem is encountered in manufacturing industries where "long life-span" goods are produced. In this case (usually because of the high initial purchase cost of the product) the product is maintained in operation for many tens of years. The problem here is that as a consequence of technological changes over time the components which make up the original product change as parts become technologically infeasible or obsolete. The result is that each individual instance of the product, with time, becomes unique. Consequently, as in the case of flexible manufacturing environments described above, the quantity of maintenance data required to provide an appropriate level of after sales services becomes increasingly difficult to again: (a) provide and (b) update. As before this is exasperated when the manufacturer concerned is operating in global markets.

The issue of the provision and updating of maintenance information for manufacturers of the form described above is the principal concern of the Stoves project. The project name is derived from Stoves PLC, the main industrial partner, who manufactures "top of the range" domestic cookers according to individual customer requirements. Stoves is an exemplar of a industry operating in a highly flexible manufacturing environment. The project is also supported by VG-Elemental who manufactures Mass-spectrometers. This is a high cost product and consequently purchasers tend to keep the product in operation for as long as possible so as to defray the original purchase cost over as long a time period as possible. VG-Elemental report that some of their spectrometers are still in operation some 30 years after the initial purchase date. This second industrial partner is thus an exemplar of the second category of industry manufacturer above.

The Stoves project is also supported by: the DTI through the Foresight-Link initiative; Euro-Serv (based in Belfast), an exemplar of a white goods maintenance service provider; and NA Software, a commercial software house based in Liverpool.

2 MODE OF OPERATION

The aim of the Stoves project is thus to provide a mechanism whereby maintenance information can be provided in a globally effective manner in such a way that it is both current and correct. The obvious medium for the provision of the desired data is the Internet. A method of working is envisaged where by a maintenance engineer located somewhere across the globe keys in a product code (what ever the product may be — cookers, mass-spectrometer, Etc.) into a WWW interface and retrieves appropriate maintenance information. This mode of operation seems to be the most obvious choice given current technology.

3 PRODUCT MODELLING

The research team do not believe that the Internet based delivery mechanism for the proposed system will present a significant challenge. The more significant issue is believed to be the storage and presentation of maintenance data. An analysis of the Stoves PLC product domain ([12]) indicates that maintenance data tends to be coupled to either individual categories of components or sub-assemblies of the product range. The first stage in the production of the desired system is therefore seen as the production of a model of the product range to which can be attached appropriate maintenance information.

Given that the requirement for this model is as a vehicle for accessing maintenance data the model need not be particularly precise — indeed it is believed that a largely qualitative model will suffice.

The most straight-forward, "brute force", approach to modelling the product range is to create a qualitative model for each version of the product. However, this has the obvious draw back that the up-keep of this data will be labour intensive. In other words the brute force approach can be said to be an identical solution to that currently provided, except that it is electronic rather than paper based. This approach is therefore seen as offering no particular advantage.

Given that there is significant overlap between product components the above also seems extremely wasteful. A more efficient mechanism would be to produce a unified data structure, which reflects the entire product range. It is also intuitively clear that, given a sufficiently high level of abstraction, a given range of products can be considered to be identical (i.e they are all cookers or mass-spectrometers). A hierarchical data structure therefore suggests it self where by each level in the hierarchy represents a different level of product abstraction. Such data structures have been used successfully in other industries for example in the modelling of chemical plants ([1]).

Further consideration indicates that a *tree* structure would be the most appropriate data structure to adopt. The nodes in this structure would represent product components with the root node representing the product at its highest level of abstraction (it may be that, given a particular type of product, it is appropriate to have more than one root node). Each node would have one or more (in most cases at least 2) nodes branching from it. A set of sibling nodes emanating from a single parent may represent a conjunction of components or a set of alternatives (disjunction). The most detailed level of decomposition would then be represented by the leaf nodes. It may also be appropriate to include *abstract nodes* in the structure that represent a logical grouping of components under a single parent node although there is no actual component of the form represented by that node. The connectors in this tree structure would then represent qualitative links of the form "is connected to" or "is part of". There is no reason why a particular node should not be connected to more than one parent (a component may be "connected to" or be "a part of" more than one other component).

Figure 1 gives a fragment of a hierarchical data structure of the form envisaged. The fragment can be used to model a range of 9 different styles of cooker. The root node represents the entire product range. The second level gives three high level components of the cooker. Note that where the arcs linking nodes are themselves connected by an arc this indicates a conjunction. In the figure, as we descend the levels of the model the level of abstraction decreases and instead of conjunctions we tend to find disjunctions as alternatives start to appear. The highlighted arcs in the Figure describe (for illustrative purposes) a cooker which has a type 1 mounted fan with an A component where the heat source is electric. Note also that the same structure can be used to model any other product range (mass-spectrometers Etc.). (Another objective of the Stoves project is to produce a solution which is as generic as possible.)

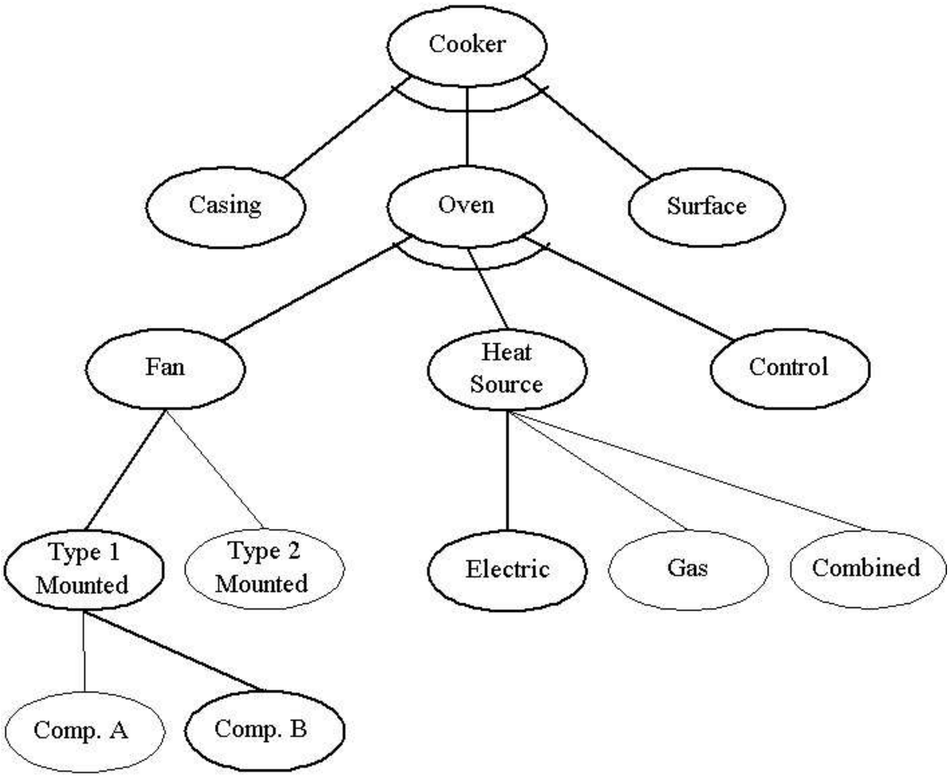

Figure 1: *Hierarchical tree data structure for product modelling*

4 PROVISION OF MAINTENANCE DATA

The modelling of the product range is seen as the first step to the desired provision of maintenance data. Conceptually, the part of a product represented by each node may have maintenance data associated with it. This data can have two purposes: (a) fault identification and diagnosis, (b) fault repair; the first precedes the second. We thus conceive of one or more *information providers* to be attached to each node each of which is concerned with either fault finding or fault rectification. The nature of these providers is diverse:

1. Canned text.
2. Dynamic text generators.
3. Canned plans
4. Dynamic plan generators.
5. Photographs.
6. Video clips.
7. Animations.
8. Databases (DBs).
9. Virtual Reality (VR) "fly throughs".
10. Intelligent VR agents.
11. Production rule systems.
12. Model based reasoning systems.

13. Case Based Reasoning (CBR) systems.
14. Other.

The first nine of the above provide predominantly information concerning restorative action after a fault has been identified; the following four are essentially used for diagnostic purposes. The last, "other", has been included to reflect further techniques which may become appropriate. Different *information providers* can thus be attached to different nodes in the hierarchy (Figure 2), and may be either:

private Accessible from only the node to which it is attached.

public Accessible from anywhere in the structure

protected Accessible from only the node to which it is attached and its sub-nodes.

This categorisation is seen as a mechanism for preventing information overload, it is not intended as a mechanism for preventing end users from accessing information. Thus (with respect to Figure 2) a VR sequence on how to dismantle a fan may be applicable to both Type 1 and Type 2 fans so should be attached to the `fan` node as a protected information provider which can be *inherited* by all the nodes descending from this node. A fault finding Qualitative Reasoning (QR) system for Type 2 fans would be particular to that node and would thus be a private information provider attached to that node. Databases containing general information may be attached to the root node and made protected so that they can be accessed from all other nodes, or may be more logical attached to some other node and thus would be declared public to achieve the same end.

5 REVIEW OF INFORMATION PROVIDERS

In this final section of the paper we wish to present ideas concerning the individual information providers listed above. In particular we wish to consider the intelligent aspect of the providers and the significance with respect to the end task. We commence by considering information providers relating to corrective action and then go on to consider diagnostic systems.

5.1 Canned text

The ability to explain faults and remedial action is seen as an important facet of the proposed system. The most straight forward mechanism where by this can be achieved is to present the user with stored explanations (*canned text*). In many cases such information is already available in paper form and thus the canned text concept can be implemented efficiently and effectively. This is the approach adopted by many "help" systems, for example Unix "man" pages, and has been extended by the use of hypermedia techniques to provide a wide range of documentation ([5]).

5.2 Dynamic text generators

Dynamic generation of explanations ([8]) is an attractive option where a significant amount of component overlap exists (as is the case here), in particular it reduces the preparation effort associated with the preparation of canned text (see above). Issues to be addressed here are mostly concerned with mechanisms for "pulling together" the elements of a desired explanation.

5.3 Canned plans

In the same way that canned text can be provided we can also provide other forms of a data in this manner — collectively such data can be described as *Canned information*. As such canned information can be considered to include photographs, video clips and animations as well as plans. In the case of plans this offers the attractive advantage that in many cases this information is already available in paper format.

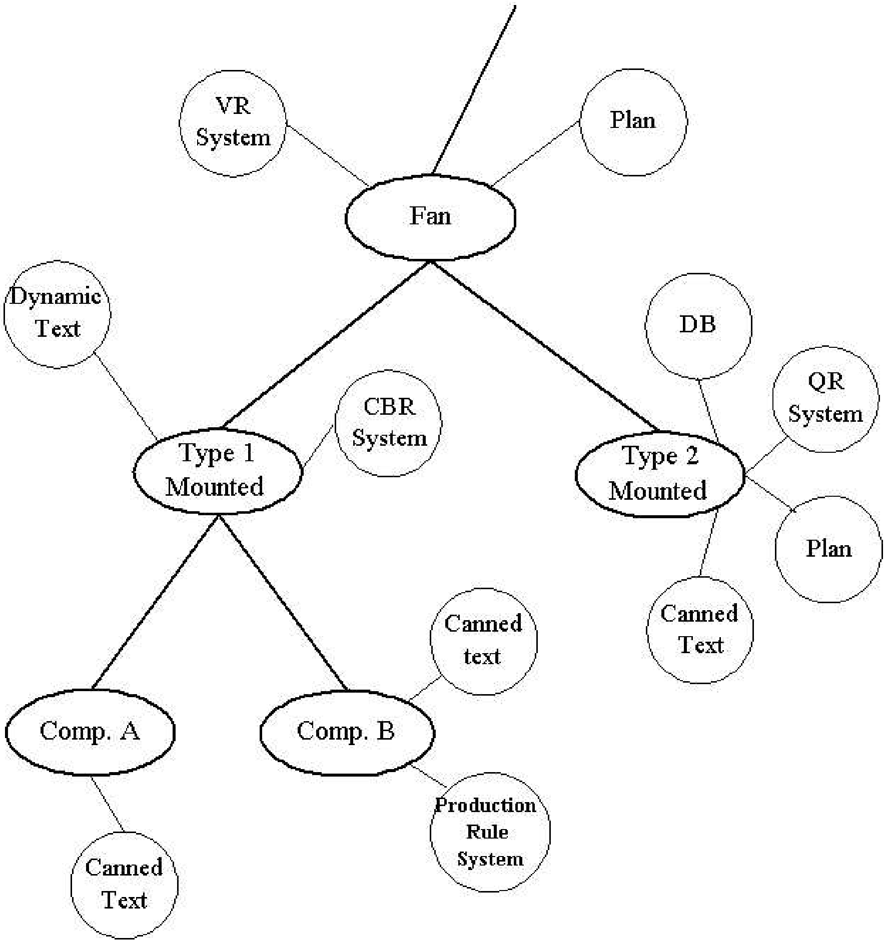

Figure 2: *Information providers attached to nodes in data structure*

5.4 Dynamic plan generators

An interesting alternative to canned plans is to generate such plans dynamically from knowledge of the individual component plans. This may involve linking into a Computer Aided Design (CAD) package.

5.5 Photographs, video clips and animations

Photographs, video clips and animations can all be considered to represent other forms of canned information. These are seen as important because, in maintenance terms, they convey information much more succinctly than text. There is of course a data storage overhead that requires reconciliation.

5.6 Databases (DBs)

An analysis of the Stoves PLC operation ([1]) indicates that a frequent requirement is the identification of part numbers. Such data can ideally be stored in a database format which can then be interacted with. The nature of these databases, with respect to certain components may be very small (say 3 fields and 10 records). In other case they may be much larger.

5.7 Virtual Reality (VR) "fly throughs"

Much work has been done using Virtual Reality (VR) Techniques, for Example, VRML ([4]) and Java3D ([9]). The former is about building 3D content on the Internet and the latter is an application programming interface (API) for drawing 3D graphics using the Java language. The project aims at applying VR techniques to the generation of product 3D models from part 3D graphics. At Stoves, there exist part 3D models created by the R&D department. Those models could be exported in different standard formats. Therefore, the main issue is to determine all structure variables and generate product 3D models following the structure rules.

5.8 Intelligent VR agents

Intelligent VR agents (IVRA) ([1]) adds intelligence to virtual reality and forms relevant autonomous agents. One of the most important applications of IVRA is to add intelligence so as to model some physical behaviours. The aim here is to match Expert Systems with VR interfaces, for example, model-based diagnosis and 3D product models. Therefore, intelligence can guide the use of VR, and VR can exploit the results of intelligence.

5.9 Production rule systems

Production rules systems have been favoured by the Expert System community for many years ([3]). They lend themselves to fault diagnostics because this task is often considered in terms of an "if-then-else" set of rules. The topic of *rule-based* fault diagnostic has therefore been a research field in the Expert Systems community for many years (for example [7]). Again there is no need for these production rule systems to be very sophisticated , a rule-base comprising some 10 rules may be perfectly adequate with respect to a particular component (node in the tree).

5.10 Model based reasoning systems

Model-based reasoning ([6]) is a diagnostic mechanism where by a model is manipulated by appropriate reasoning routines. For example, one classic approach is to use a "perfect" model to generate the ideal operation, then introduce faults into the model to reproduce the behaviour of the real world system. When the model's behaviour matches the actual behaviour, the faults in the model are candidates for the real world faults. Model-based systems have, in recent years, found favour with industry, for example in the Tiger project ([10]). The drawback of the technique is that minimal (or no) "explanation" is offered.

5.11 Case Based Reasoning (CBR) systems

Case based reasoning ([11]) is another commonly used tool, both for diagnostics and to produce corrective information. The theory is based on the observation that practitioners, in what ever field, derive their expertise from previous occurrences of situations (cases). In CBR these cases are stored in some form of data structure fronted by a reasoning systems. A new case is then presented to the system which then searches through its data base looking for similar cases to the new case. Information is then returned concerning these previous cases which, it is argued, can then be usefully employed to resolve the new case. It would seem that in the Stoves project CBR would be another useful tool to have available with respect to particular nodes (components). Again a CBR system need not be particularly sophisticated, a lot may be done with a history of only (say) 100 previous cases.

5.12 Other

Whatever information providers are adopted with respect to the Stoves project it is clear that the data structure must also make provision for the addition of new information providers including those that have yet to be conceived. Similarly the data structure must also allow for the adaptation, perfection and correction of information providers; and if necessary there removal.

6 CONCLUSION

In this paper we have described the DTI Foresight-Link supported Stoves project which commenced in December 1999 within the Department of Computer Science at the University of Liverpool. The central aim of the work is to produce a demonstration maintenance support system founded on a hierarchical tree structure which is used to model the product, the nodes of which have various information providers attached.

The work has only been in progress for a short period of time and the research team have only recently completed the "domain analysis" phase of the work([1]). The proposals outlined here are founded on this initial analysis. The research team have now commenced on the modelling phase of the work.

7 ACKNOWLEDGEMENT

The research is supported by the industrial partners mentioned above. In particular we would like to give our thanks to: Bob Weaver, Dave Rothwell, Dave Crozier and Jamie Alsamarraie from Stoves PLC; Stephen Gilmore, Seamus Lively and Gerry McGivern from Euro-Serv; Mike Delves from NA Software Ltd; Allan Smith and Stan Price from DTI; and Ian Fitch from the Connect Center at the University of Liverpool.

References

[1] Aylett, R., Petley, P., Chung, P.W.H., Soutter, J. and Rushton, A. (1999). Generating Operating Procedures for Chemical Process Plants. Integrated Manufacturing Systems - The International Journal of Manufacturing Technology Management. v10, n6, p328-42.

[2] Ballin, D. and Aylett, R. (1999) Intelligent Virtual Environment and Virtual Agents. Tutorial at the 19th SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence. 13 Dec. 1999.

[3] Buchanan, B.G. and Shortliffe, E.H. (1984). Rule Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley.

[4] Carey, R., Bell, G. and Marrin C. (1997). The Virtual Reality Modeling Language (VRML97). ISO/IEC 14772-1. The VRML Consortium Incorporated.

[5] Crowder, R.M., Wills G., Heath I. and Hall, W. (1997). The Application of Hypermedia in the Factory Information Environment. Proc 5th Int. Conf. on FACTORY 2000, Cambridge, April 1997; IEE Conference Publication 435.

[6] Hunt, J.E. (1996). Model Based and Qualitative Reasoning: Industrial Application, 1st International Workshop on Model-Based and Qualitative Reasoning — Perspectives for Industrial Applications, ECAI '96, Budapest.

[7] Moulton, M. (1997). A Rule-Based Incident Tracking System. In Macintosh, A. and Milne, R. (Eds), Applications and Innovations in Expert Systems V, Proceedings ES'97, SGES publications, London.

[8] Rubinoff, R. (1986) Adapting Mumble: Experiences with Natural Language Generation, Proceedings AAAI-86.

[9] Sowiztal, H., Rushforth, K. and Deening, M. (1998). The Java 3D API Specification. Addison-Wesley.

[10] Milne, R., Nicol, C., Travé-Massuyès, L., Quevedo, J., Ghallab, M., Bousson, K., Dousson, C., Aguilar, J. and Guasch, A. (1994). TIGER: Real Time Situation Assessment of Dynamic Systems, Intelligent Systems Engineering Jo., 3, No. 3.

[11] Watson, I. (1997). Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufman Publishers, London.

[12] Zhang, W and Weaver, B. (2000). Stoves Product Domain Analysis. Stoves Project Deliverable SP_D1.1. Feb. 2000, Department of Computer Science, The University of Liverpool.