Reasoning by Activation-Based Evidence Propagation

Jacob L. Cybulski
Amdahl Australian Intelligent Tools Programme
Department of Comp. Sci. and Comp. Eng.
La Trobe University
Bundoora, Vic 3083, Australia
Fax: +613 470 4915

Rick Coxhill and Chris D. Rowles
Artificial Intelligence Systems
Telecom Research Laboratories
770 Blackburn Road Clayton, Vic 3168, Australia
Fax: +613 543 8863

Abstract

This paper describes a model of reasoning by evidence propagation. The model combines symbolic and connectionist paradigms of representation and processing, and was implemented in NEXPERT and C. Its test-bed application was in the area of preventive maintenance, resulting in a prototypical expert system predicting failures occurring in a telephone network and arising from the penetration of water into the network components, e.g. cables and joints.

1 Background

EXPRES (EXternal plant PReventative maintenance Expert System) is being developed to aid the process of predicting telephone network faults arising from the penetration of water into cables and joints. The initial presence of water in the network components causes only slight deviation in resistance and external voltage measured at cable pairs serving individual clients (incipient faults). As the water damage spreads through the network, the quality of network services slowly deteriorates until such time as the condition is no further tolerable by the client and the cost of repair becomes significantly elevated. EXPRES was, thus, designed to detect early signs of water penetration and to predict the most likely locations of cable and joint damage so that the network maintenance could be planned ahead and performed at the time convenient to the network provider. (cf. Richardson 1985 for applications of AI to maintenance.)

EXPRES project went through several prototypical phases in search for the best modeling and implementation frameworks. First, it was implemented in Personal Consultant Plus and DBase III. Then, the need for the system portability across different hardware platforms forced the research into the use of Unix-based software. The new expert system shell was selected based on its efficiency and the quality of its user interface, thus, Personal Consultant was replaced with NEXPERT Object (henceforth referred to as NEXPERT). This paper discusses an early prototype of the EXPRES system implemented in NEXPERT and C and utilising a novel, activation based, evidence propagation (cf. Quillian 1968, Collins and Loftus 1975). To differentiate this experimental version from the current development system, which utilises other methods of evidence manipulation, we will call it μEXPRES.

NEXPERT allows μEXPRES knowledge representation in an object-oriented formalism, permits interfacing to the existing databases via a set of standard primitives, provides a rule-based inference engine, offers a foreign function interface for a non-standard object and rule manipulation or provision of graphical interfaces, etc.

The first stage of μEXPRES analysis is to construct an accurate representation of the customer access network in NEXPERT (ref. Figure 1); i.e., each network device is classified into one of several categories, then each of the objects is described by a number of its class-specific attribute-values obtained from external databases, next the addition of inter-object connections allows to approximate a true network topology, and finally, to increase the system run-time efficiency, data resident in objects is compressed, summarised, and a number of statistics are computed. The reasoning stage of μEXPRES starts with importing the results of measuring pair resistance and foreign battery voltage. This initiates the use of domain rules and evidence propagation mechanism to locate those network components which are most likely to fault in the near future.

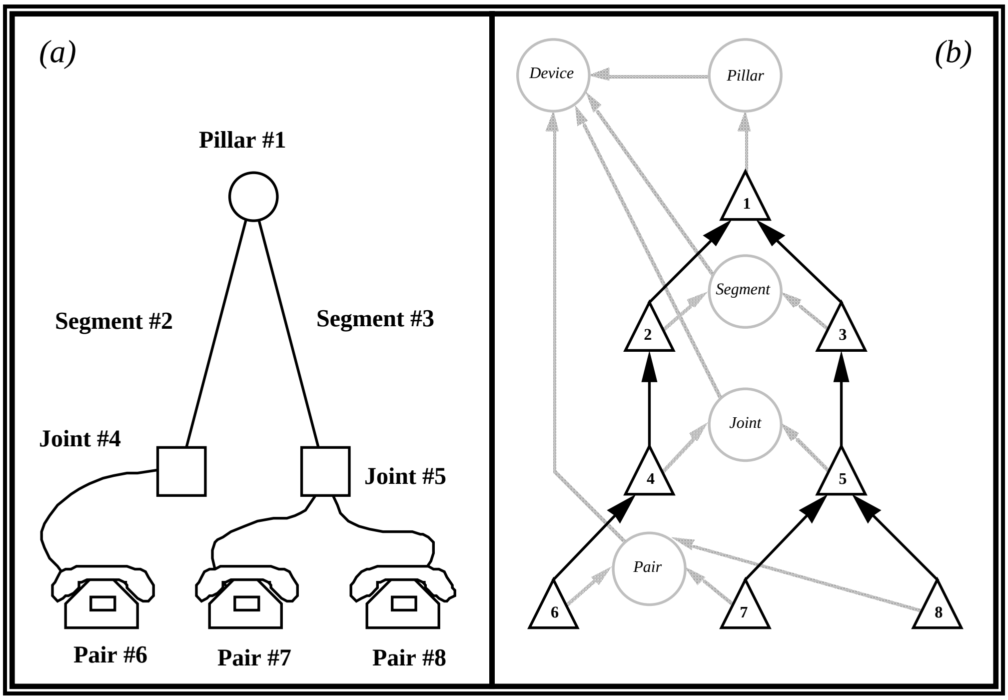

Figure 1 - customer access network (a) and its representation (b)

2 Reasoning and Representation in μEXPRES

From the very beginning of the μEXPRES project, it was quite evident that a large volume of knowledge about network operation and repair is already in existence and used by network maintenance engineers. The knowledge proved to be easily extracted from experts and formalised into a body of predicate rules, a rule system was hence a preferred platform for the μEXPRES inference engine (Hayes-Roth 1985). At the same time, a description of all telephone network plant was found to be well organised and described in appropriate technical documentation, manuals, and manufacturer reports. So, it became apparent that of all different knowledge representation schemata, it was an object-oriented model (Minsky 1975), allowing taxonomic description of network devices, that would be most advantageous to the project for its concise and elegant form. After the new framework for fault prediction was constructed, it was realised that the system lacks a human-like ability to fine-tune its expert knowledge base depending on the history of incipient fault occurrence, repair and the subsequent maintenance. A form of neural-like fault likelihood propagation and training (Lippman 1987, also cf. Zaidenberg 1990) was, thus, proposed to alleviate the problem, this also aided the rule selection, activation and firing. Let us describe the three levels of representation and manipulation in μEXPRES, i.e. the knowledge representation, evidence propagation and inference rules.

Knowledge Representation. All network *devices* (NEXPERT objects) may be classified into four distinct categories (NEXPERT classes), namely :- *pillars*, large joints, serving all cable pairs in a district area, cable *segments* carrying a number of pairs from the pillar to the customer sites, *joints* interconnecting, splitting or adjoining the cable segments, and finally cable *pairs* allowing direct connection of telecommunication services to the client's equipment. Each of the device types is characterised by a number of inheritable attributes, e.g. pillars - pillar identifier, type, size, used and total capacity, distance from the exchange, pair range; joints are similar to pillars - joint identifier, type, size, used and total capacity, distance from the exchange, pair range, housing type, pit size, equipment type and its size, and the construction type; segments are characterised by the following attributes - segment identifier, the types of cable conductor, bearer, sheath, installation, use, appearance, pair, and transmission, bearer size, pair range, feed designation and its direction, reference to the exchange and country side joint, finally the exchange side cable segment; and finally pairs by - their identifier, associated telephone number (if allocated), feed cable segment identifier, its resistance and voltage, feed designation and distribution point identifier, finally the pair condition rating. Further classification and attribution of pillars, joints, segments and pairs may be considered to increase efficiency of the reasoning tasks, at this stage, however, the rough classification into four main categories is quite adequate.

The second issue in μEXPRES knowledge organization is the representation of network topology. Telephone networks are predominantly hierarchical, therefore, individual network components may be arranged in a similar fashion as well, occasionally found ill-formed network structures may be mapped into a pure hierarchy, by splitting the loops and circuits or by duplication of split-ends, without a great loss of domain information. NEXPERT permits object structures to be independent from their class taxonomy, certain irregularities in NEXPERT inheritance procedure, however, forced μEXPRES to adopt a different strategy controlled externally from a C program. Each device contains a *Parent* slot indicating the plant closer to the pillar, and thus, defining a network hierarchy. The slot is used by the rule system to reason about inter-device connection, it is also used by the hierarchical evidence propagation sub-system to spread device fault likelihoods. Additional *Weight* slot is set to the value proportional to the number of pairs going through the device, it, thus, reflects the statistical aspects of the network hierarchy and is being used to weigh the fault evidence coming from the lower levels of the hierarchy. Small changes in the distribution of factors allocated to the *Weight* slots, and motivated by the influx of current and historical maintenance data, may be used to adjust the evidence propagation paths.

Evidence Propagation. μEXPRES reasoning attempts to trace and justify aberrations in resistance and voltage registered at the client distribution pairs and measured from the exchange by a special SLIQ test. Such measurements may indicate problems occurring not only at the client end of the network but frequently relate to the condition of all devices installed on the path from the district-pillar through a number of network components down to the individual client site.The role of the expert system is to combine the body of evidence about a network topology, the type of network devices, pair measurements, and historical information of the network condition and repair, all to mimic a possible propagation of incipient faults from individual pairs up to the pillar as it occurs during the SLIQ test. A fault propagation analysis, thus, is controlled by a number of domain rules and is supported by an activation based evidence propagation mechanism (using a number of specialised slots - Figure 2).

The concept of evidence propagation follows the mechanism of activation spreading in hierarchical hypothesis spaces (cf. Shyu 1989) and neural networks (cf. activation spreading, e.g. Quillian 1968, Collins and Loftus 1975, Williams 1986). The fault

Parent	*Constructs a network topology*
Weight	*Defines the connection weight*
Thres	*Specifies the fault likelihood threshold of device*
Up	*The likelihood of fault residing above net device*
Fault	*Likelihood of device being at fault*
Down	*Fault determined by sub-devices*
In	*Initial fault likelihood at pairs*
SigExp	*Determined internally by EXPRES*
SigRcv	*Used internally by EXPRES*

Figure 2 - Fault propagation slots

presence is detected at pairs (*In* slot) and being propagated up to the higher levels of representation (along the *Parent* hierarchy using the slots *Up*, *Fault*, *Down*, *Weight* and *Thres*). Whenever a network component is seen as faulty from the pillar's vantage point (*Down* > *Thres*), it is assessed what proportion of fault may be contributed to the component behaviour (*Fault* slot) and what is the possibility of devices above its level to give a faulty reading (*Up* slot). Once the fault evidence is calculated for all devices having a common *Parent*, their faults (from *Up*) are weighted (i.e. multiplied by *Weight*) and accumulated (in *Down*, i.e. $Down = \Sigma Weight_i \infty Up_i$).

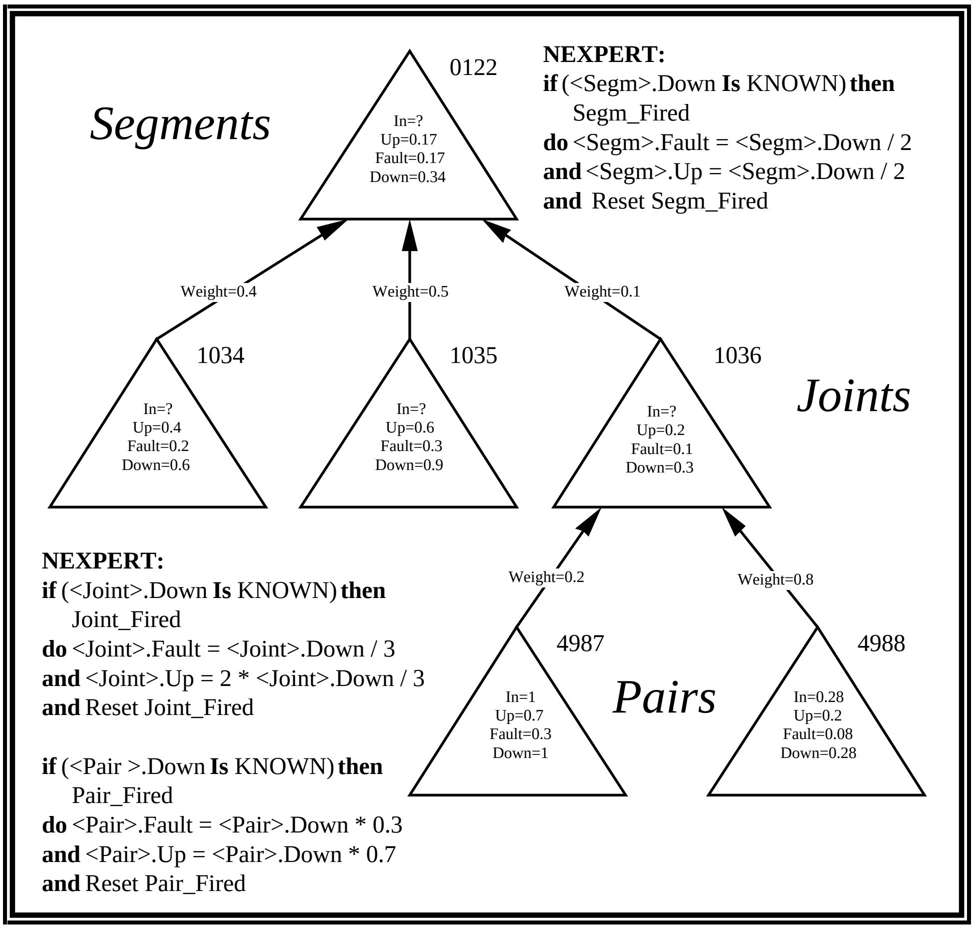

Figure 3 - Evidence propagation between network layers

Consider a simple example illustrated in Fig. 3. Let us investigate two faulty pairs 4987 and 4988. The SLIQ test determined them to give a 100% faulty reading, thus *In*=1. The assessment rules for pairs (in this simple example) indicate the 0.7/0.3 split between the *Fault* and *Up* fault likelihood. The *Up* faults are weighted and accumulated in the *Down* slot of a higher network component, joint 1036. The process then alternates between joints and segments until the fault assessment of the district pillar is completed. At this point we may look back at all network devices and select the critical ones having the highest fault rating (here • 0.3, at joint 1035 and pair 4987).

μEXPRES evidence propagation mechanism is organised as a collection of 9 external C functions, of which some allow the manipulation of NEXPERT slots, some support the *Parent* hierarchy, others control the evidence propagation between *In*, *Up* and *Down* slots. Three of the above-mentioned functions are installed as NEXPERT 'handlers', thus, making them available for μEXPRES reasoning tasks (propagating evidence from *Up* to *Down*, initiating the evidence propagation from pairs' *In* up to *Down*, and finally initialising the *Parent* hierarchy by setting the internally used synchronisation slots *SigRcv* and *SigExp*).

Inference Rules. There are three different types of inference rules used by μEXPRES at different stages of its execution (Fig. 4): *compression rules* preparing the network global statistics necessary for the system subsequent use, *propagation rules* triggered automatically by the propagation mechanism, and localised to individual network devices assessing their fault likelihood, and finally *selection rules* which given the overall network condition could localise the areas of fault development.

1. Start μEXPRES*(evdprop + cable.tkb)*

2. Define network device attributes (external data base)
3. Define network topology (external data base, set Parent, SigExp)
4. Define all SLIQ test results (external data base, set In)

5. Compress domain knowledge (Compression rules: e.g. Pair-Range stats)
6. Propagate evidence (Propagation rules: Up, Down, Fault, Thres, Weight)
7. Localise faults (Selection rules: Fault > Global Threshold)

8. Report and quit μEXPRES

Figure 4 - Eight stages of μEXPRES execution

The reasoning core of the μEXPRES system is a set of propagation rules. Their role, as illustrated in Fig. 3, is to detect a device threshold condition (e.g. *Down* > *Thres*, or availability of the fault evidence), and to respond to it with the provision of an assessment procedure dividing the fault evidence coming from the bottom of the network hierarchy (*Down* slot, i.e. total fault evidence at this or higher levels) into a component remaining with the device being assessed (*Fault* slot, i.e. the fault is caused by the device itself), and the portion belonging to the higher levels of the hierarchy (*Up* slot, i.e. the fault is caused by devices closer to the pillar). The rule structure and their logic, although very simple, when combined with their sheer volume and the evidence propagation, gives the system immeasurable inferencing power.

a. ***if*** *a cable is UG* ***then*** *suspect lead-in joints above lead-in ($\bar{U}$p)*
b. ***if*** *IR • 50K* ***then*** *100% chance of failure in 2 months ($\bar{F}$ault)*
c. ***if*** *a joint contains •100 pairs* ***then*** *it is unlikely to fail (¬Fault)*
d. ***if*** *a joint distributes only 1 faulty pair* ***then*** *the fault is in this joint ($\bar{F}$ault)*
e. ***if*** *•67% of segment pairs are faulty* ***then*** *suspect the segment ($\bar{F}$ault)*

Figure 5 - μEXPRES sample rules

Fig. 3 shows a set of very simple propagation rules only, their actions are well defined and localised to individual objects. Although such rules are possible in μEXPRES application (e.g. Fig. 5 a-c), majority of rules do refer to attribute-values of neighbouring objects in the network (e.g. Fig. 5 d), and some to distant areas of network hierarchy (e.g. Fig. 5 e). To prevent constant scanning of the network for characteristic devices and their properties, some pre-processing of data by compression rules is necessary (e.g. fault statistics about faulty pair-ranges going through segments and joints may have to be calculated).

Once the propagation of evidence is completed, a number of network devices are marked as probably at fault or likely to fault in future. A number of selection rules are applied to choose for reporting only those devices that are critical to the network operation.

3 Conclusions

The paper discussed an evidence propagation architecture for a working system μEXPRES implemented in NEXPERT and C. The system combines elements of rule and object representation, and reasoning aided by spreading activation. Activation-based control of rule firing allowed a more modular and compact definition of rules, it also added ability to fine-tune the system reasoning by redistribution of evidence weights.

Future research on μEXPRES-like evidence propagation will concentrate on the application of the technique to other areas of fault diagnosis. Investigation of fully automatic and adaptive techniques for weights adjustment is also perceived (e.g. by backpropagation of weight faults).

4 Acknowledgements

We acknowledge the permission of Executive General Manager of Telecom Australia Research Laboratories to publish this paper.

5 References

Collins, A.M. and Loftus, E.F. (1975): "A Spreading Activation Theory of Semantic Processing," *Psychological Review*, V72, N6, 407-428.

Hayes-Roth, F. (Sept 1985): "Rule-Based Systems," *CACM*, V28, 921-932.

Lippmann, R.P. (April 1987): "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, IEEE, pp 4-22.

Minsky, M. (1975): "A Framework for Representing Knowledge," in P.H. Winston (ed) *The Psychology of Computer Vision*, New York, NY: McGraw-Hill, 211-280.

Quillian, M.R. (1968): "Semantic Memory," in Minsky, M.L. (ed) *Semantic Information Processing*, Cambridge, MA: The MIT Press.

Richardson, J.J. (Ed. 1985): *Artificial Intelligence in Maintenance*, Park Ridge, N.J.: Noyes Pubs.

Shyu, C.-H. (1989): "Heuristic Evidential Reasoning in a Hierarchical Hypothesis Space," in M.H. Hamza (ed) *Expert Systems and Neural Networks: Theory and Applications*, Anaheim: ACTA Press.

Williams, R.J. (1986): "The logic of Activation Functions," in Rumelhart, D.E. and McClelland, J.L. and the PDP Research Group (eds) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: The MIT Press, V1, 423-443.

Zaidenberg, M. (1990): *Neural Network Models in Artificial Intelligence*, New York: Ellis Horwood.