Transactions on Information and Communications Technologies vol 20, © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

An architecture for knowledge discovery in complex telecommunication systems

R. Sterritt, K. Adamson, C.M. Shapcott, D.A. Bell
School of Information and Software Engineering,
Faculty of Informatics, University of Ulster,
Newtownabbey, Co. Antrim, Northern Ireland.
EMail:{R.Sterritt, K.Adamson, CM.Shapcott, DA.Bell}
@ulst.ac.uk

Abstract

Applying Artificial Intelligence (A.I.) to complex engineering problems is an increasing trend. The authors have been involved in researching and applying such AI techniques as data mining, induction, genetic algorithms, probabilistic nets, machine learning, to a complex non-deterministic engineering problem in the telecommunications domain. This has been implemented as a Knowledge Discovery architecture (NetExtract) which is generic in nature.

1 Introduction

This paper discusses the knowledge discovery architecture developed. First the telecommunications domain is described, followed by an overview of NetExtract and probabilistic nets - which are used to describe the complex relationships discovered. The architecture is then discussed in terms of each component.

1.1 Telecommunications Systems

Within the telecommunication industry the Synchronous Digital Hierarchy is an international standard for broadband networks, offering increased bandwidth and sophisticated services(CCITT[1]). This increased sophistication allows traditional voice, video on demand, ISDN data transfer and video conferencing to use the one network more efficiently and effectively.

The management of this level of sophistication becomes more difficult, particularly when a fault in the network occurs(CCITT[2]). The SDH multiplexers themselves and other network elements (NE) have built in recovery methods and the behaviour of these NE are highly specified by ITU (CCITT), ANSI and ETSI and as such are deterministic.

When a fault does occur lower level components within a multiplexer generate alarms. Further alarms are then generated up the network hierarchy as components at that level also detect a fault. After a time period masking does take place, but by this stage large amounts of alarm data have been generated.

An added feature is that each multiplexer is responsible for relaying alarms generated downstream to the network manager, effectively cascading the large amount of data. For example, a single fault on a multiplexer can generate up to 6Mb of binary alarm data making it difficult to isolate the true cause of the fault. The net effect of this complexity is that the behaviour of the multiplexer network is effectively non-deterministic(Bouloutas et al[3]).

In simple terms this project aimed to take a birds eye view of the behaviour and extract cause and effect networks from it, as opposed to a bottom up approach which analyses the effects of each and every possible state of a multiplexer in relation to the network in which ever state it happens to be in.

The NetExtract approach can be generally described as knowledge discovery; a mix of data mining (to extract meaningful knowledge from these alarms) and evidential reasoning (to describe the behaviour in terms of causes and effects or to describe the relationship between evidence (alarms) and hypotheses of interest (faults)).

1.2 NetExtract - An Architecture for the Extraction of Cause and Effect Networks from Complex Systems

The initial work had two parallel strands. In the first strand the team evaluated an existing parallel database prototype (Bell et al[4]) and produced a design for needed extensions. The second strand consisted of an examination of various approaches to evidential reasoning, including those using probabilistic networks (Buntine[5]) and the Dempster-Shafer[6] theory of evidence. An overall design for the NetExtract architecture was proposed (Moore et al[7]).

The team evaluated several such algorithms to be used in inducing Bayesian nets. Two algorithms were prototyped, OMI (optimisation of mutual information), and a genetic algorithm (Cause and Effect Genetic Adaptation algorithm - CAEGA). A research tool for extracting chain graphs from data, BIFROST (Højsgaard et al[8]) was also evaluated. All three gave similar results when used with the test data.

The proposal of the NETEXTRACT architecture, in common with most machine learning approaches, uses a relation-based approach to the data. Information concerning the behaviour of test cases is assumed to be available and is passed, possibly after suitable pre-processing, to an induction process, which extracts a model - in our case a Bayesian net - from the data.

The induction process attempts to optimise the net – in the sense that it searches for the net structure that best fits the data. The Bayesian net thus extracted can subsequently be used as an expert system.

1.3 Probabilistic Nets

Nets in which relationships between variables can be represented by the existence of links between them have an intuitive appeal. They are easy to “read” if represented graphically and can summarise fairly complex relationships succinctly. Probabilistic nets are a special case of such nets (others include decision trees and neural networks). A probabilistic network defines graphical relationships which express various independence properties possessed by the variables. Simon[9] has shown that if the variables can be regarded as structural parameters in some sort of mechanism then a probabilistic net in which variables are connected in a directed acyclic graph can be created which represents the behaviour of the mechanism.

As a very simple example, assume that we have a database containing information about the co-occurrences of the following three events: X1: a drop in the outside temperature from warm to near-freezing, X2: the occurrence of heavy rain, X3: a flip in the central heating thermostat. If we know that X1 has occurred then it changes our expectations of X2, and X3, but we don’t expect that X3 will depend on X2 directly. It is intuitive to express this relationship as a directed graph with X1 as the parent of X2 and X3. The joint probability distribution for the three variables, X1, X2 and X3, can be represented by the product of conditional probability tables, one table for each variable. The table specifies the conditional probability that the variable takes each of its possible values given the values taken by its parents. In the simple case in point the joint distribution takes the form:

$$Pr(X1, X2, X3) = Pr(X 2 \mid X 1) \times Pr(X 3 \mid X 1) \times Pr(X 1) \qquad (1)$$

This example can be readily generalised to more complex nets involving more variables and many other examples in which “causes” and “effects” can be linked in this way have been used to good effect in the statistical and artificial intelligence literature. In their classic paper Lauritzen and Spiegelhalter[10] illustrate how such nets can be used as expert systems. The prime advantage of having such a representation is that it reduces the computational effort in using Bayes’ law for diagnosis. The brute force effort involved in computing the conditional probability of unobserved variables, given the known values of observed variables is cut down considerably by judicious use of the known properties of the structure.

In many cases, when generating Probabilistic Nets, the structure of the net is not known in advance, but there is a database of information concerning the frequencies of occurrence of combinations of different variable values. In such a case the problem is that of induction – to induce the structure from the data. Unfortunately the general problem is NP-complete. For a given number of variables there is a very large number of potential graphical structures which can be induced. The only sure way of determining the best one is to fit the data to each possible graphical structure, score the structure, and then select the structure with the best score. Consequently algorithms for learning networks from data are always heuristic.

2. The Architecture

2.1 Overview

Although the exemplar was from the telecommunications industry, the research, design and development targets a generic architecture that would be equally suitable for medical diagnostics, financial forecasting/prediction and geological surveying etc..

Due to that generic nature, the development was undertaken in a “bread-board” fashion. The architecture can be characterised by four main components (Figure 1):

❑ Data Cleaner
❑ Pre-Processor
❑ Induction
❑ Deduction

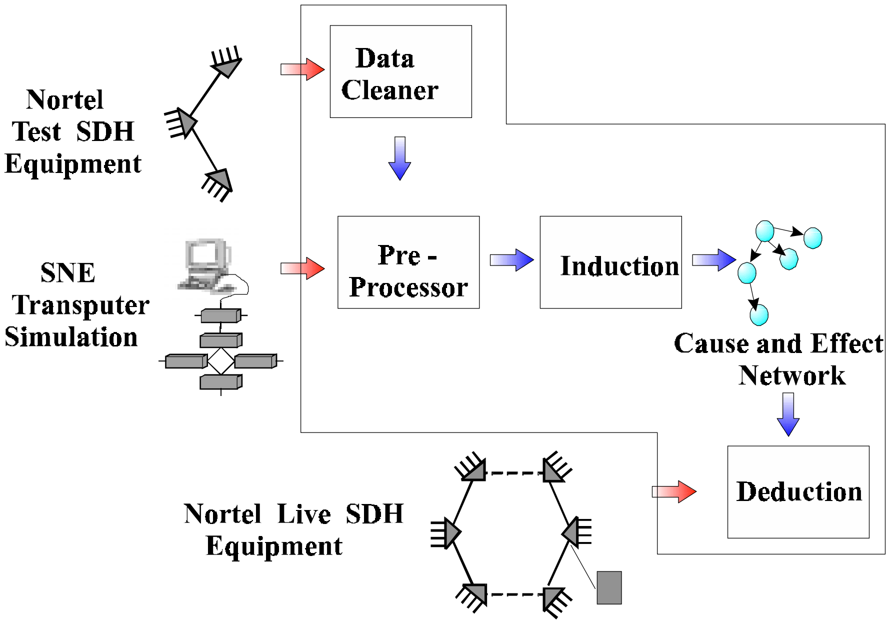
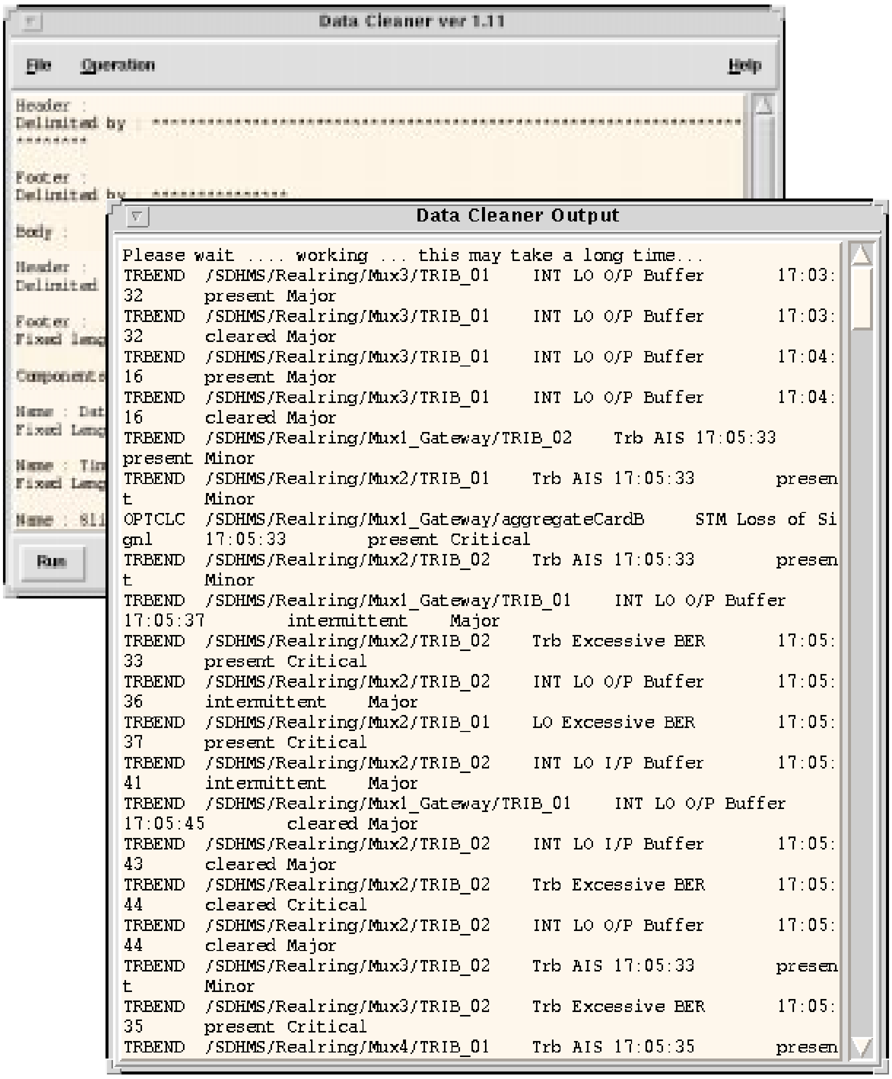

Nortel Test SDH Equipment
SNE Transputer Simulation
Nortel Live SDH Equipment
Data Cleaner
Pre - Processor
Induction
Cause and Effect Network
Deduction

Figure 1 NetExtract Architecture

The first three components work with the data to produce the cause and effect networks/graphs or Bayesian Belief Networks. They find the hidden evidence and produce a model of cause and effect for the domain. The deduction component then uses this network as a model upon which to predict or diagnose faults from live data.

There are three sources of data in the exemplar. The first being the SDH network at BNR NITEC (Northern Ireland Telecommunications Engineering Centre) where new components are tested. The second is our joint development, the Synchronous Network Emulator (SNE) which currently simulates limited behaviour of a SDH network (the development is being carried out on INMOS transputers on another project). The third is potentially live data from the field. Once a cause and effect network that models the behaviour accurately has been produced (with additional input from SDH experts) the model (cause and effect network) could be used with the deduction component to diagnose faults in live data. The cause and effect network would only have to be re-modelled when a change in the design of SDH's occurred that caused new or different behaviour.



```
Header :
Delimited by : **********************************************************************
********

Footer :
Delimited by : ***************

Body :

Header :
Delimited

Footer :
Fixed leng

Components

Name : Dat
Fixed Leng

Name : Tim
Fixed Leng

Name : Sli
```

```
Please wait .... working ... this may take a long time...
TRBEND  /SDHMS/Realring/Mux3/TRIB_01    INT LO O/P Buffer      17:03:
32      present Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_01    INT LO O/P Buffer      17:03:
32      cleared Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_01    INT LO O/P Buffer      17:04:
16      present Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_01    INT LO O/P Buffer      17:04:
16      cleared Major
TRBEND  /SDHMS/Realring/Mux1_Gateway/TRIB_02    Trb AIS 17:05:33
present Minor
TRBEND  /SDHMS/Realring/Mux2/TRIB_01    Trb AIS 17:05:33       presen
t       Minor
OPTCLC  /SDHMS/Realring/Mux1_Gateway/aggregateCardB     STM Loss of Si
gnl     17:05:33        present Critical
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    Trb AIS 17:05:33       presen
t       Minor
TRBEND  /SDHMS/Realring/Mux1_Gateway/TRIB_01    INT LO O/P Buffer
17:05:37        intermittent    Major
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    Trb Excessive BER      17:05:
33      present Critical
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    INT LO O/P Buffer      17:05:
36      intermittent    Major
TRBEND  /SDHMS/Realring/Mux2/TRIB_01    LO Excessive BER       17:05:
37      present Critical
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    INT LO I/P Buffer      17:05:
41      intermittent    Major
TRBEND  /SDHMS/Realring/Mux1_Gateway/TRIB_01    INT LO O/P Buffer
17:05:45        cleared Major
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    INT LO I/P Buffer      17:05:
43      cleared Major
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    Trb Excessive BER      17:05:
44      cleared Critical
TRBEND  /SDHMS/Realring/Mux2/TRIB_02    INT LO O/P Buffer      17:05:
44      cleared Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_02    Trb AIS 17:05:33       presen
t       Minor
TRBEND  /SDHMS/Realring/Mux3/TRIB_02    Trb Excessive BER      17:05:
35      present Critical
TRBEND  /SDHMS/Realring/Mux4/TRIB_01    Trb AIS 17:05:35       presen
```

Figure 2 Data Cleaner Screen Shot

2.2 Data Cleaner

Data cleaning is often an understated part of the discovery process that in certain domains can take up to 80% of the total time (Hatonen[11]). This component (Figure 2) converts the verbose raw data (Figure 3) into easily manipulated database tuples (Figure 4) discarding the redundant data. It is generic since the user can create a template from the data description language (Figure 5), which indicates what information is to be taken from the raw data.

```
Event Message Receive
___________
Date : 02/12/1996
Time : 17:03:34
Slip Number : 614
Affected Object Type
Affected Object Path
/SDHMS/Realring/Mux3/
Event Type : INT LO O
The Event occurred at
02/12/1996 17:03:32
Alarm Status : presen
Severity : Major

Event Message Receive
___________
Date : 02/12/1996
Time : 17:03:38
Slip Number : 615
Affected Object Type : TRBEND
Affected Object Path :
/SDHMS/Realring/Mux3/TRIB_01
Event Type : INT LO O/P Buffer
The Event occurred at :
02/12/1996 17:03:32
Alarm Status : cleared
Severity : Major

Event Message Received :-
___________
Date : 02/12/1996
Time : 17:04:18
Slip Number : 616
Affected Object Type
Affected Object Path
/SDHMS/Realring/Mux3/
Event Type : INT LO O
The Event occurred at
02/12/1996 17:04:16
...
...
...
```

Figure 3 Sample of the Data from Nortel Test SDH Network

```
TRBEND  /SDHMS/Realring/Mux3/TRIB_01  INT LO
O/P Buffer     17:03:32       present Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_01  INT LO
O/P Buffer     17:03:32       cleared Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_01  INT LO
O/P Buffer     17:04:16       present Major
TRBEND  /SDHMS/Realring/Mux3/TRIB_01  INT LO
O/P Buffer     17:04:16       cleared Major
TRBEND  /SDHMS/Realring/Mux1_Gateway/TRIB_02
        Trb AIS 17:05:33      present Minor
TRBEND  /SDHMS/Realring/Mux2/TRIB_01  Trb AIS
        17:05:33       present Minor
...
...
...
```

Figure 4 Part of the corresponding output from the Data Cleaner

```
Header :
Delimited by :
***********************************************
**************************

Footer :
Delimited by : ***************

Body :

Header :
Delimited by : ___________

Footer :
Fixed length : 1

Components :

Name : Date
Fixed Length : 19

Name : Time
Fixed Length : 17
...
...
```

Figure 5 Part of the corresponding data description file to describe the format of the test data

2.3 Pre-processor

The cleaned data is then sifted through by the pre-processor as part of the knowledge discovery process. Firstly it produces an observation table either by observing the entire history or by specifying a time-frame. In our exemplar the observation variables are the alarms which can have a binary value (observed or not observed). In actual fact the alarms themselves can have three states present, intermittent or cleared. An alarm is observed in a time frame if it is present or intermittent. It will be included in the following time-frames until a cleared message is encountered for that alarm. This adds a temporal dimension to the discovery process.

At this stage the user may use the observation table to perform induction by using such algorithms as BIFROST.

The observation table is further refined by creating a contingency table. A contingency table, in simple terms, is a list of various variable combinations that have occurred in the history observed, and their frequency of occurrence.

The contingency table is then used with such algorithms as CAEGA and OMI to produce a cause and effect network (induction).

The next stage in the knowledge discovery process is performed by the novel approach of combining Cooper and Herskovits[12] and our Genetic Algorithm to score the structures according to the database of observations which takes place in the Induction component.

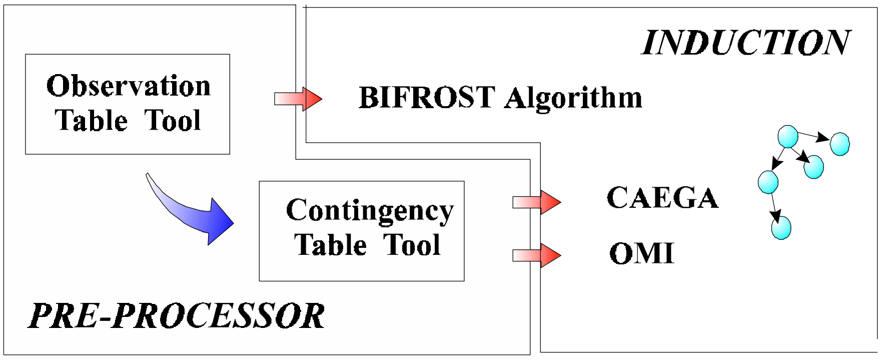


Figure 6 Interface between Pre-processor and Induction

2.4 Induction

The output from the knowledge discovery process then provides input for the various evidential reasoning algorithms we tested. From these initial results it was decided to concentrate on our generic genetic algorithm (CAEGA) to produce a cause and effect network.

Genetic algorithms are based on genetics, the science of heredity, which involves studying the structure and functions of genes and the way they are passed from one generation to the next. The difference between organisms are the result of differences in genes they carry, differences which have resulted from breeding:

- ❑ mutation (a heritable change in the genetic material)
- ❑ crossover (exchange of genetic materials between chromosomes)
- ❑ selection (favouring of particular combinations of genes in a given environment)

The algorithm searches for a probabilistic structure (which states the connections between the variables), that has a high probability given the contingency table.

For a specified number of generations the algorithm provides mutation by allowing random changing of limited elements in some structures at each generation. Breeding takes place by exchanging equivalent elements in two structures (the parents) every generation. Selection occurs by favouring the structures that score the best against the contingency table. The weaker structures are not ruled out since they may contain important elements for the gene pool that would not be advisable to exclude too early.

2.5 Parallel Induction

The project has a parallel dimension. As well as the initial parallel database investigation and the SNE simulation running on transputers it was decided to implement an induction algorithm in parallel for knowledge discovery, due to the sheer volume of data involved in data mining (Agrawal et al[13]).

The obvious candidate was CAEGA due to the time required to execute and the intrinsic parallel nature of genetic algorithms (Bertoni[14]). Like the SNE simulation the implementation was carried out on INMOS T805 transputers connected to a Sun workstation.

The initial version consists of a straightforward master - slave implementation (processor farm). The breeding (reproduction, crossover and mutation) could be carried out in parallel. We also implemented the scoring and scaling in parallel but the communications cost in transmitting these back to the master removed any benefit from just having the master perform these functions. The selection had to be implemented sequentially and thus remained on the slave. This is simply because all the structures from the new generation require to be re-mixed to form new parents from the gene pool before being distributed to the slaves for breeding (Sterritt[15]).

As expected the scalability is limited because of the overhead of communications. This is not a problem in the SNE since the scalability simply involves an increase in the number of Network Elements of a specified network topology modelled.

This implementation (P-CAEGA) can be classified as global parallelisation. Every individual has a chance to mate with all the rest (i.e. random breeding), thus this implementation did not affect the behaviour of the original algorithm (CAEGA).

In the longer term we plan to investigate and develop a more sophisticated parallel approach where the population is divided into sub-populations, relatively isolated from each other. This model introduces a migration element that would be used to send some individuals from one sub-population to another. This adapted algorithm would yield local parallelism and each transputer could be thought of as a continent or a country were the majority of the breeding occurs between residents with limited migration.

This paradigm would then be feasible to implement on other parallel architectures for instance using PVM.

2.6 Deduction

The output from the induction algorithm is a Bayesian Belief Network. The deduction module acts as an expert system, using the Bayesian network as the vehicle for answering "if then" questions. Input to the deduction module is either a network generated by the induction algorithm, or an externally defined network. The module will display the network to the user who is then able to explore the effects of changing variable values. For example, if Alarm type Y is observed, this alters the probability that alarm Z will be observed.

To increase the generic nature of the architecture the Bayesian Network Interchange Format (BNIF) from Microsoft Research[16] and the UAI (Uncertainty & Artificial Intelligence) community has been implemented to store the networks.

2.7 GUI

A graphical user interface (GUI) has almost become the defacto interface to any application today. This has occurred mainly because of the proliferation of the PC into the hands of the traditionally computer unfamiliar.

Although the architecture was designed to be generic, the main driving force was the telecommunications exemplar. Nortel in fact did not require a GUI as part of the architecture. This was due to the fact that their engineers, experienced computer workstation users and developers, preferred full screen style interfaces which allow quick manipulation. Another reason was that since the process of producing a cause and effect network from large amounts of data can be a lengthy one, they would prefer the facility of writing scripts to automate the calling of components, enabling the process to be left running overnight. Also the first three stages do not require interaction once execution begins.

This created a conflict since most other users now expect a GUI. Our compromise was to develop a limited GUI that at least provides the naive user with the correct usage of the architecture. With more experience they would be able to proceed to writing scripts. This GUI has been implemented using TCL/TK from Sun and C. The deduction component which requires manipulation of nodes in the graph and thus requires interaction will be further developed with a GUI.

3. Conclusion

3.1 Evaluation

The project produced a useful research study into the possibility of a novel Knowledge Discovery architecture for such a complex problem using various AI techniques (data mining, induction, genetic algorithms, probabilistic nets).

This architecture has been implemented in a generic fashion. The system runs on a Sun SparcStation with a network of T805 transputers connected. The initial results look promising from the test data supplied by NITEC.

To complete the KD process requires “interruption and evaluation”, the human element. It is only now that NITEC are in the position to build up a data mountain to utilise the architecture, they are near completion on a automated testing facility of the SDH test network.

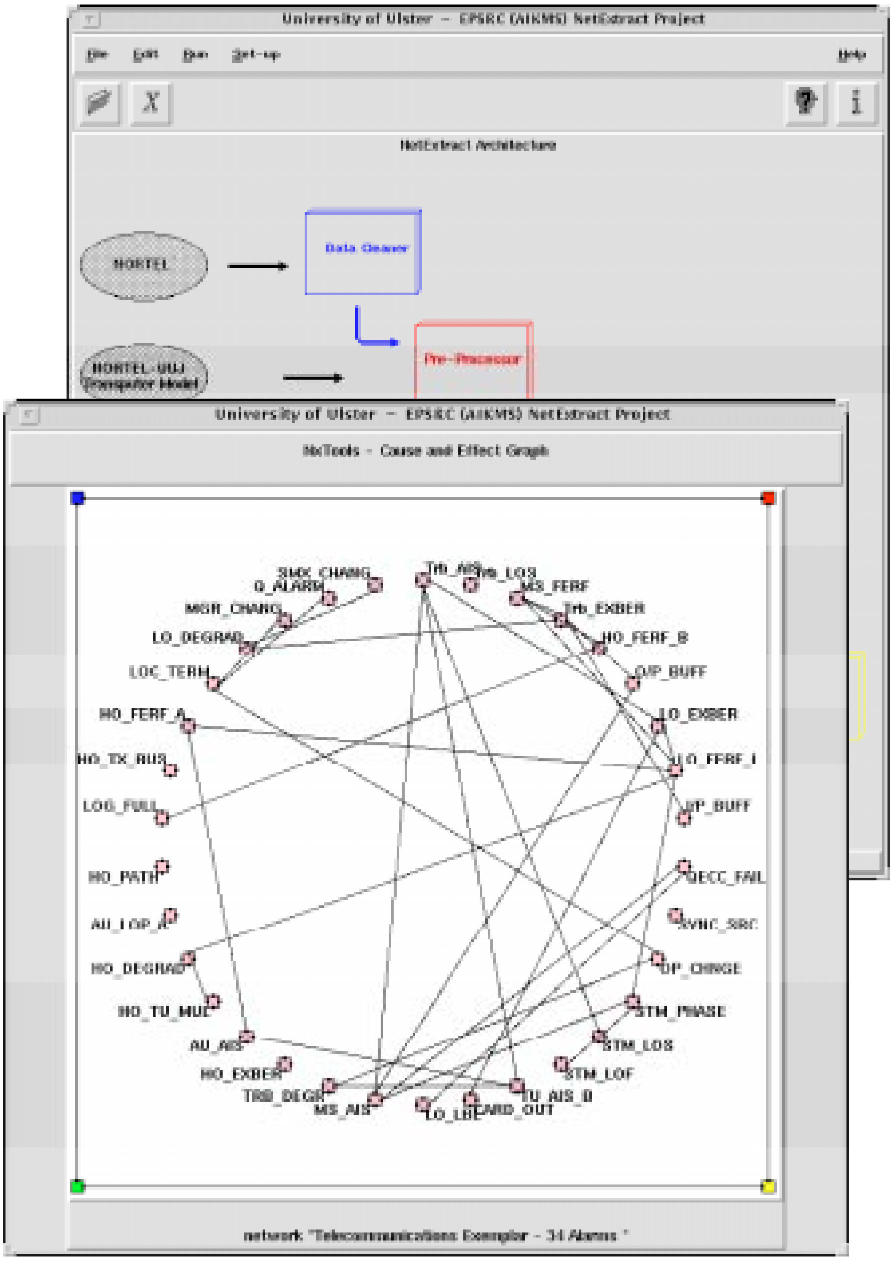


Figure 7 Screen Shot of NxTools[17]

3.2 Future Work

The team will monitor the application of the architecture to automated data at NITEC, ensuring correct function. In parallel, work continues on the other collaborative project with NITEC (developing the mentioned SNE). They also plan to carry out further development on the genetic algorithm (P-CAEGA), and also extending the user interface for the deduction component.

Acknowledgments

We would like to thank EPSRC (AIKMS programme) for funding this work and our industrial collaborators BNR NITEC (R&D facility for NORTEL), and Transtech Parallel Systems Ltd. for their support of this work.

References

[1] CCITT, *Types and General Characteristics of SDH Multiplexing Equipment*, CCITT Recommendation G.782 1990.

[2] CCITT, *SDH Management*, CCITT Recommendation G.784 1990.

[3] Bouloutas, A. T., Calo, S. and Finkel, A., Alarm Correlation and Fault Identification in Communication Networks, *IEEE Transactions on Communication*, Vol. 42, No 2/3/4, Feb/Mar/Apr 1994.

[4] Bell D.A., Hull M.E.C., Cai F.F., Guthrie D., Shapcott C.M., A Flexible Parallel Database System Using Transputers, Proc. *IEEE Int. Conf. On Computer Systems and Software Engineering*, 1992

[5] Buntine, W., A Guide to the literature on learning probabilistic networks from Data, *IEEE Transactions on Knowledge and Data Engineering*, 8(2), 195-210, 1996.

[6] Shafer G., Logan R., Implementing Dempster's rule for Hierarchical Evidence, *Artificial Intelligence* 33, pp271-298, 1987

[7] Moore P., Shao J., Adamson K., Hull MEC., Bell DA., Shapcott M., An Architecture For Modelling Non-Deterministic Systems Using Bayesian

Belief Networks, *Proc. of the 14th Int. IASTED Conf. On Applied Informatics*, pp254-257, 1996

[8] Højsgaard S.,Thiesson B., BIFROST - Block Recursive Models Induced From Relevant Knowledge, Observations and Statistical Techniques., *Computational Statistics and Data Analysis*, 19, pp155-175, 1995

[9] Druzdel M.J., Simon H.A., Causality in Bayesian Belief Networks, *Proc. of 9th Conf. On Uncertainity in AI*, pp3-11, 1993

[10] Lauritzen S.L., and Spiegelhalter D.J., Local Computations with Probabilities on Graphical Structures and their Applications to Expert Systems, *J. Roy. Strat. Soc. B*, 1988

[11] Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, P., Toivionen, H., Knowledge Discovery from Telecommunication Network Alarm Databases, *12th ICDE* 1996.

[12] Cooper, G. F. and Herskovits, E., A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning*, 9, pp309-347, 1992.

[13] Agrawal, R., Imielinski, T., Swami, A., Database Mining: a Performance Perspective, *IEEE Transactions on Knowledge and Data Engineering*, 5(6), pp 914-925, December 1993.

[14] Bertoni A., Dorigo M., Implicit Parallelism in Genetic Algorithms, *Artificial Intelligence* (61) 2, pp307-314.

[15] Sterritt R., Adamson K., Shapcott M., Wells N., Bell D.A., Lui W., P-CAEGA: A Parallel Genetic Algorithm For Cause And Effect Networks, *Proc Int. IASTED Conf. On AI and Soft Computing*, pp105-108, 1997

[16] Microsoft Decision Theory and Adaptive Systems Group, Proposal for a Bayesian Network Interchange Format, *Microsoft Discussion Paper*, 1996

[17] Sterritt R., Adamson K., Shapcott M., Bell D.A., McErlean F. , Using A.I. For The Analysis Of Complex Systems, *Proc. of the Int. IASTED Conf. On AI and Soft Computing*, pp105-108, 1997