Automatic Search for Correlated Alarms

Klaus-Dieter Tuchs, Peter Tondl, Markus Radimirsch, Klaus Jobmann
Institut für Allgemeine Nachrichtentechnik, Universität Hannover
Appelstraße 9a , 30167 Hanover, Germany
Tel.: +49-511-762-2836, Fax: -3030
e-mail: tuchs@ant.uni-hannover.de

Abstract: *The main topic of this paper is fault management, especially the search for correlated alarms in large alarm records stored in databases. We have chosen a GSM/DCS mobile telephone network as a basis for the investigations of data mining algorithms which are used to detect correlated alarm patterns. Special attention has been paid to the access network. The main problems addressed are the evaluation of alarm bursts, the task of alarm correlation and the development of tools to evaluate those bursts. These correlation tools need information for the correlation of alarms which today are only known to particularly educated and experienced system experts. These experts are expensive and the task itself is also extremely expensive and time-consuming. Moreover, they are not able to find all correlations that would appear in a large network. The fallibility of human operators was the motivation to develop a data mining tool that is able to find correlated alarms with the aim to minimise the costs of the alarm evaluation process.*

1 Environment of the research work

Modern communication networks consist of a huge number of network elements that are interconnected with each other. It is their co-operation that enables the provision of network services to subscribers.

Operation of a communication network requires that alarm messages can be displayed and evaluated. These messages are generated by single network elements but are, gathered at a central point, an indication of the current status of the network. The messages considered in this article are mostly error and alarm messages. They are used for the recovery from errors and to monitor the current performance of the network. This is particularly important for early detection of congestion situations which can be caused by errors. Moreover they are the basis for manual corrections of the network structure and parameters by the operator.

A big variety of error situations exists where a single event causes a huge amount of error messages. Such error burst are e.g. generated when one or more radio links between network elements in a GSM network fail due to a thunderstorm. These failures usually cause a series of other network elements to drop out as well and each drop out results in at least another one, very often a number of error messages. Investigations have shown that the drop out of a single radio link may result in up to 300 subsequent error messages which all arrive almost simultaneously at a central network operation and maintenance centre.

The ideal solution in such situations with respect to a quick error recovery would be a system which is able to derive a single and unambiguous indication of the original error reason. Moreover, this system should be able to generate a detailed list of necessary actions to clear the failure.

Recent developments in fault management exhibit a trend to introduce systems which reduce the number of displayed alarms. Such systems use error correlation techniques. Their task is to evaluate a huge amount of error messages that contain redundant information and semantic repetitions and, thus, complement each other. The goal is to identify a potential original reason at the beginning of the error chain, taking stochastic properties of the error propagation into account. This evaluation can be based upon neuronal networks [1], model based artificial intelligence systems [2], [3] or simpler rule-based methods.

These methods lead to a broader decision basis for the operator which increases the efficiency of the network in operation.

2 Automatic search for alarm patterns

The alarm correlation systems described in clause 1 have in common that they require background informations about the alarm messages they shall correlate. Today, these background informations about alarm patterns that are specific for certain error situations are obtained from the network by experienced experts by means of observation and evaluation of a number of identical error situations. After a certain time, these experts have learned to filter the alarm messages such that they can detect the originating error, even if the number of displayed errors is huge. This learning process is very time consuming and costly because it needs to be repeated frequently due to changes and extensions of the network topology and software updates. The consequence is that the error patterns change significantly which requires costly manual updates and maintenance of correlation systems.

It can easily be recognised that, for the given reasons, a simple correlation system is not sufficient but a system is required that is able to automatically update the background information for the actual correlation. The task of such a system would be the automatic search for correlation patterns. This method is generally known as data mining and is the main subject of this paper.

The data bases applied in today's communication networks enable the realisation of systems whose purpose is to search for typical patterns, e.g. in error data bases. This paper presents a data mining algorithm that is able to find alarm patterns out of huge amounts of data which can be traced back to a single error reason. The basics for data mining have been introduced in e.g. [5] where such algorithms are applied to problems in big supermarkets or department stores.

3 Prerequisites for the application of data mining algorithms in network management

The alarm messages generated in a communication network are usually stored in an alarm data base. Each alarm is represented by predefined attributes, cf. [4].

alarmidentifier	networkelement identifier	alarmnumber	eventtime	other attributes...

Figure 1: Format of an alarm database entry

Figure 1 shows the attributes of a stored alarm message that are relevant for the data mining algorithm.

The number of entries in the above described fault management data base can reach enormous numbers, depending on the size and type of the network under consideration.

The goal of an automatic data mining algorithm is to detect periodically appearing combinations of alarm messages within the alarm data base which are potentially caused by a single reason. These alarm combinations have to be assessed by the algorithm according to certain guidelines in order to decide whether a typical and, thus, valid error pattern has been detected.

An error pattern shall be denoted as valid if it contains only alarms that are caused by a single error source and are, therefore, indirect errors. Such an error source is e.g. the failure of a single radio link [7].

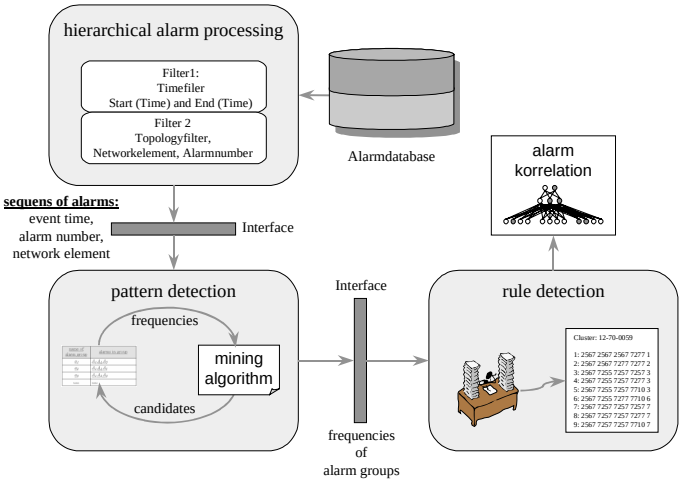

Figure 2: Structure of the Data Mining Tool

The structure of the operation and maintenance support tool for correlated alarms is shown in figure 2. It is divided into three independent parts:

1. Preprocessing of the alarm events based on hierarchical network information
2. Pattern detection
3. Rule construction

The upper part in Figure 2 shows the filter functions that reduce the number of investigated alarms. The mining function is the subject of this paper. It is responsible for the search of correlated alarms and is depicted in the lower part. A detailed description of its functions will follow in the following clauses.

3.1 Hierarchical alarm processing

For the reduction of alarms to be investigated, the filter function uses the topology of the network. The criterion is the search for groups of network elements that are physically or logically connected with each other and are therefore able to generate error series.

The first step is the analysis of the network with respect to its hierarchical structure. For this purpose, the algorithm needs to have access to the topology data base. The result of the analysis is a hierarchical topology model of the network. An example for such a model is shown in Figure 3. The left graph describes the model that is generated from the real network. This model represents the GSM/DCS access network that is shown in the right graph.

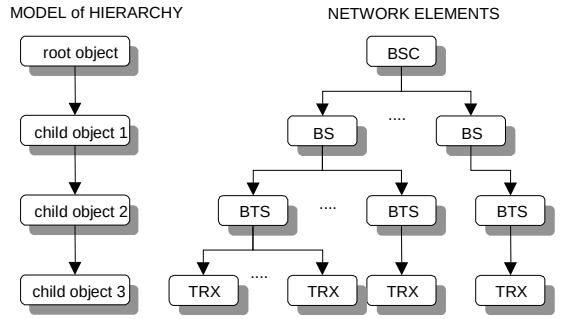

Figure 3 GSM hierarchy model

The hierarchical model presented is an abstraction which only considers dependencies directly related to the hierarchical structure of the real network.

In figure 3, the hierarchical tree has two branches below the root. In this example, an error occurring in one of these branches can never cause any error messages in the other branch, i.e. the branches are independent from each other. The pattern search can, therefore, be performed independently for each branch.

In the remainder of this paper, the root was placed on Base Station (BS) level, i.e. only the elements below the BS level were considered (see figure 3).

A refinement of the model is planned in the future. However, it has turned out that good results can be achieved already with this simple model.

3.2 Pattern detection

The algorithm used was introduced in [8] and has been transformed and optimised by the authors for the usage in radio communication networks.

For the purpose of describing the algorithm, we define *S* as the set of alarm events to be investigated. The algorithm subject to this paper only considers the alarm type and the event time of each alarm event, as shown in figure 1.

The pattern search algorithm needs to know all possible alarm types, A_k, where $k \in \{1,2, ..., M\}$ and *M* the maximum number of known alarm types. Then, $S = \{a_i\}$, $i = 1, 2, ..., N$, with $a_i \in \{A_k: k \in \{1, 2, ..., M\}\}$.

We furthermore assume that an order function is defined for the alarm types such that, if $i>k$, then $A_i \geq A_k$. We now define a *group* as a set of alarms containing *L* alarm types in ascending order with respect to their alarm identifier. The ordering ensures that each combination of alarm types occurs only once in the set of all possible groups of length *L*. Due to the order function of the alarm types, all groups with equal length *L* can also be arranged in ascending order. A group of length *L*, then, is denoted by F_L^p where *p* represents the position in the sequence of all groups of length *L*. Then, $F_L^p = \{a_1, a_2, ..., a_L\}$. Table 1 shows a selection of groups for $L=3$ and $M=5$.

Name of alarm group	Alarms in group
F_3^3	$A_1\ A_1\ A_3$
F_3^5	$A_1\ A_1\ A_5$
F_3^8	$A_1\ A_2\ A_3$
.....	.....

Table 1: Some groups for $L=3$ and $M=5$

Due to the ascending order with respect to the alarm identifiers, all alarm groups with the same alarms in their first element, a_1, are following each other. E.g. if the first *L*-1 alarms of the alarm groups F_L^i and F_L^j with length *L* are identical, these alarms are a subset of all possible F_L^p for all *p* with $i \leq p \leq j$.

This subset of alarm groups F_L^p for all *p* with $i \leq p \leq j$ is called a *block*. Potentially interesting alarm groups are called *candidates*. Candidates of length *L*+1 can be found by building all possible combinations of alarm pairs within a block F_L^p for all *p* with $i \leq p \leq j$.

The set *S* is passed to the pattern detection algorithm such that all events a_i are arranged in ascending order with respect to their event time. We call *S* also a *sequence* of events. Hence, the event time defines a *time axis* onto which the alarm events are mapped.

The goal of the Data Mining algorithm is to analyse this *time axis* of events and to find frequently appearing groups of events that occur with a certain probability and that have identical or similar properties.

The detection of the alarm groups in the *sequence* is described in the following:

Frequency search:

The frequency search is based on a sliding window scheme. Note that the window is defined in time units, i.e. the numbers of alarms in the window at different times may vary. The duration of the window is defined as T_M. The window is shifted in steps where a step is equal to shifting the window by one alarm event.

In the first step, the search algorithm counts events that correspond to groups of length $L=1$, i.e. each group consists of a single event A_k and the maximal possible number of groups corresponds to *M*. For each existing alarm group a counter counts the number of alarm events in the current window of duration T_M. The window is shifted step by step until it reaches the end of *S*. Note that a single event can be counted several times, e.g. if the window contains more than one alarms when evaluating this event.

After *S* has been evaluated, the resulting counters are regarded for each group. If the counter value exceeds

a certain limit, these alarm types are kept for further investigation. If the counter value resides below this limit, the alarm type is removed.
In the second step, the length of the groups is extended to L=2. The groups, however, contain only those alarm types that have passed the previous round, i.e. they have not been removed. All possible groups are build and arranged in ascending order with respect to their alarm identifier. Note that the set of groups is not anymore complete with respect to the set of possible alarm types due to the removal step at the end of the first round. The resulting set of groups consists of the blocks with those A_k that have survived the first round.

The sliding window search is now applied to S a second time. For each allowed group exists a counter which counts the occurrence of the group in the window. A group is said to occur if the two alarm types of this group appear within the window. This appearance does not depend on order or position of the alarm types. Note that a group can be counted more than once, especially if the window contains the group in more than one step.

Again, the counters are evaluated. If a group has occurred more often than a certain limit, it survives. Otherwise, it is removed.

At the beginning of each step, a surviving group of length L builds the basis for a new allowed block of groups with length (L+1). The extension works as follows, see also example in figure 4:

- Among the surviving groups of length L in step L, those are selected that build a block of length (L-1).
- The alarm types a_L of these survivors are considered to be allowed extension for the next – step for all new blocks of length (L+1) with the previously considered block of length (L-1) as a basis.

Let {A, B, C, D} be the set of allowed alarm types in the example in figure 4. At the end of the third round, the (ordered!) groups {ABA}, {ABC}, {ABD}, {ACC} have survived. We get the basic groups {AB} and {AC} of length 2 for further processing. The alarm types of block {AB} have the alarm types A, C and D as their last element. Each allowed group of the block {AB} that has survived round three is now extended by these last elements of the survivors. This is explained in figure 4. The group {ACC} is extended in the same way.

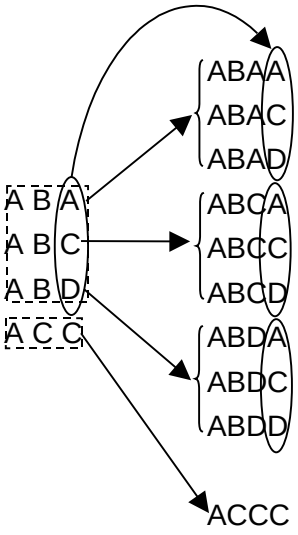

Figure 4: Example for building new allowed groups

This mechanism is iterated until the algorithm reaches a termination criterion. In our investigations, this criterion means that no new frequent groups can be found.

3.3 Rule detection

After finishing the pattern search the results have to be examined. The algorithm detects groups of different length. Table 2 shows a result from an example Base Station Subsystem (BSS). The *sequence* of alarms given to the algorithm had a length of 82 Days. The column 'BS' shows the identifier of the examined Base Station. The next column describes the number of alarms which were generated from the BS and the subordinated network elements (see figure 3). Finally the last columns represents the number of alarm groups found by the data mining algorithm. E. g. the area of the BS 56 had generated 190 frequent alarm events during the 82 days. Six different frequent groups of alarms with the length 5 were found.

BS	Alarms	L =				
		1	2	3	4	5
49	175	6	9	6	2	1
51	116	3	5	5	4	1
55	2964	10	19	11		
59	190	4	14	22	21	6
69	278	4	3			
70	1675	7	13	9	1	
72	142	8	24	12	2	
93	565	7	15	5	1	
103	145	5	11	10	1	

Table 2: examble for groups of frequencies

The rule detection has to find uncorrelated groups from alarms of different length. The difficulty here is that alarm patterns that do have e.g. a length of 4

have a high count in round 4 and, therefore, survive and go into round 5. Since there is no alarm pattern with the length of L=5 starting from this alarm pattern, this branch of the tree dies after round 5. The rule search algorithm, however, needs to be able to sort out these alarm patterns as well.

The principal way to proceed is to evaluate the resulting tree in backward direction. The basic goal of this work is to find all independent single groups of different lengths which represent a valid alarm pattern with high probability.

The development of the rule detection is for further study and has just started.

4 Summary and outlook

The algorithm presented is able to detect alarm patterns on the basis of the topology model and present it to an operator for examination. The only task left to the operator is to relate the alarm pattern to a single error reason.

Work is going on to optimise the parameters of the mining algorithm for the search of correlated alarms. One of the most important criteria is the decision whether a alarm group a_i survives or not. The criterion currently used is an absolute threshold for which, according to current investigations, it is hard to find sensible values. It will be investigated whether dependencies exist to the overall length of the alarm sequence, N, the window size T_M and the way the alarms are distributed in the time window.

It should be noted that the three functional blocks shown in figure 2 can be treated independently from each other. Hence, the work done for the data mining algorithm can be used seamlessly in the already present functions for the rule detection and filtering with the hierarchical model.

5 References

[1] H. Wietgrefe, K.-D. Tuchs, K. Jobmann, et.al: “Using Neural Networks for Alarm Correlation in Cellular Phone Systems”, International Workshop on Applications of Neural Networks to Telecommunications 1997 (IWANNT’97), Melbourne, Australia June 1997

[2] P. Fröhlich, K. Jobmann, W. Nejdl: “Model-Based Alarm Correlation in Cellular Phone Networks”, International Symposium on Modelling Analysis and Simulation of Computers and Telecommunication Systems MASCOTS 97, Haifa, Israel, January 1997, pp. 197-204

[3] P. Fröhlich: DRUM-II: “Efficient Model-based Diagnosis of Technical Systems”, dissertation on the University of Hanover 1998

[4] ETSI Technical Specification TS 100 251: “Digital Cellular Telecommunication System: Fault management of the Base Station Subsystem” (GSM12.11); July 1998

[5] V. Ganti, J. Gehrke, R. Ramakrishnan: "Mining very large Databases", IEEE Computer Magazine August 1999, pp.39-44

[6] Hugo Navas: “Design und Implementierung eines Data-Mining Algorithus für das Fault-Management in Mobilfunknetzen”; Masterthesis, University of Hanover, Institute for Communications 1998

[7] M. Cech: “Evaluation der Anwendungsmöglichkeiten von Data Mining-Algorithmen im Netzmanagement von Mobilfunknetzen”, diploma thesis, University of Hanover, Institute for Communications, 1996

[8] Hannu Toivonen: "Discovery of Frequent Patterns in Large Data Collections"; University of Helsinki; ISBN 951-45-7531-8 1996