FAST ALARM PROCESSING WITHOUT CONNECTIVITY INFORMATION

Adriano C. LISBOA
ENACOM – Brazil
adriano.lisboa@enacom.com.br

Ezequiel C. Pereira
CEMIG D – Brazil
ezequiel.pereira@cemig.com.br

Douglas A. G. VIEIRA
ENACOM - Brazil
douglas.vieira@enacom.com.br

ABSTRACT

This paper introduces an alarm processing without using any connectivity information of the substation electrical network. In order to do so, timestamp and location readily defines specific alarm patterns. In this sense, the proposed algorithm resembles the typical procedure that operators do. This specialized alarm processing allows fast recognition of the most important events in a substation: typically, more than 70% of the alarms belong to the three groups presented in this paper. Tests in a notebook show that it is possible to process 1025 alarms per second with a state machine that model the specialist behaviour of each group, while a typical maximum density is 1,000 alarms per minute.

INTRODUCTION

CEMIG is responsible for operating over 400 substations in its operation area. A single center remotely monitors all these substations. Many fault events may occur simultaneously and the operational crew must analyze them all for readily deciding the proper way of fixing them, may it be remote or local. Moreover, it is very common that a single event generates a series of alarms in cascade, making it harder to find the causing event. Under this context, automatic alarm processors are very helpful for supporting real time data mining of operators.

Alarm processing typically searches for the key information of the cause of alarms for grouping related alarms together. For an exact inference, typically the whole substation system is modelled (i.e. components and their connectivity) [1] [2] so that an optimization formulation may be properly stated for fault location [3] [4]. This exact fault location inherits a high computational burden and may not be suitable for real time operation in large systems.

In order to improve computational performance but losing in accuracy, pattern search approaches have been proposed [5] [6] [7]. In this approach, an accurate model of the substation is usually not necessarily, which includes the connectivity of components. Instead, alarm data collected from the substation is used to train and extract patterns associated to faults. Once patterns have been identified, they may be also processed using standard algorithms [8].

This work takes a step further in the pattern recognition approach: it considers only three patterns that respond for more than 70% of the faults, it uses a specialized processing and it uses location of the devices that generate alarms to speed up the process. As a result, typically about one thousand alarms may be processed per second in a standard computer. The origins of this approach is from an attempt to mime the own procedure of expert operators.

ALARM PROCESSING

When a fault event occurs in a substation, many devices generate alarms. Besides the relevance of knowing and logging triggered alarms, grouping alarms related to the same causing event is very convenient for the operator, as shown in Figure 1. With such automation, the operator readily distinguishes concurrent fault events.

Grouping

In order to exactly process each alarm and associate it with the correct causing event, it is necessary to know all components of a substation and their connectivity. However, it is possible to process the most frequent fault events without any connectivity information and still achieve suitable results: knowing where the component lies and the event time are already enough. This strategy for alarm processing enormously simplifies the alarm processing algorithms and significantly reduces the keeping effort of input database. Furthermore, this is the typical strategy used by an operator when no automatic alarm processors are available.

This work considers the most common grouping types: automatic reclosing of circuit breakers (Figure 1), local operation mode (Figure 2) and transformer fault (Figure 3). These three groups typically cover more than 70% of fault events in a substation.

Lay-out

The text should be in two columns. Please do not change the column settings. To control column length and to keep each column even, you may insert “column breaks”. If you need to insert a figure that extends over the full page width, please create an appropriate section using “continuous section breaks” before and after the figure.

Automatic reclosing

When a circuit breaker opens the system for protection, it unchains many alarms in components in the same bay which may be grouped together by a same causing event, as shown in Figure 1. Furthermore, circuit breakers typically automatically reclose after some time in order to repower the system and, after three attempts, they stop trying. This very common alarm source may be fast and precisely detected if treated in a specialized way.

	Date	Time	Alarm
1	28/08/2013	00:00:00	succesfull 2 automatic reclosing
2	28/08/2013	00:00:00	Circuit breaker open
3	28/08/2013	00:01:00	Circuit breaker closed
4	28/08/2013	00:01:01	Circuit breaker open
5	28/08/2013	00:02:00	Circuit breaker closed

Figure 1 – Automatic reclosing group.

Local mode

When a bay that is operating in local mode, usually for testing purposes, all alarms from it typically do not need any fixing operation. In this case, every alarm from that bay may be grouped into a single alarm group that indicates the local operation, as shown in Figure 2.

	Date	Time	Alarm
1	28/08/2013	00:00:00	local mode
2	28/08/2013	00:00:00	Entering local mode
3	28/08/2013	11:00:00	Circuit breaker open
4	28/08/2013	11:01:00	Circuit breaker closed
5	29/08/2013	00:00:00	Leaving local mode

Figure 2 – Local mode group.

Transformer fault

When a transformer enter in a faulty state, it typically affects the whole substation so that any alarm in that substation close in time to that event may be grouped together. This kind of fault is typically identified by the actuation of relay 86, as shown in Figure 3.

	Date	Time	Alarm
1	28/08/2013	00:00:00	local mode
2	29/08/2013	01:00:00	transformer fault
3	29/08/2013	01:00:00	Relay 86 active
4	29/08/2013	01:00:01	Circuit breaker open

Figure 3 – Transformer fault group.

State machine

A state machine is kept for each area of the substation related to an alarm group type. This state machine is efficiently implemented by keeping data only of active alarms, so that no information of state machines at initial state are stored. There are three types of state machine, one for each alarm group type. These state machines are grouped in a global state machine and each state induces a distinct way of consuming a new alarm.

Automatic reclosing state machine

When a circuit break opens due to a fault in the network, it automatically tries to close after some time. If it closes and the fault persists, it will open again. It tries reclose up to three times. The automatic reclosing state machine, shown in Figure 4, captures this behaviour: 0, 1, 2 and 3 are states related to the number of openings and 1AR, 2AR and 3AR are states related to automatic reclosing. If a reclosing is successful (i.e. no fault anymore), the state machine returns to the starting state after waiting a time to open t_o (typically 30 seconds). Conversely, if the reclosing does not happen after some waiting time t_c (typically 3 minutes), the state machine returns to the starting state.

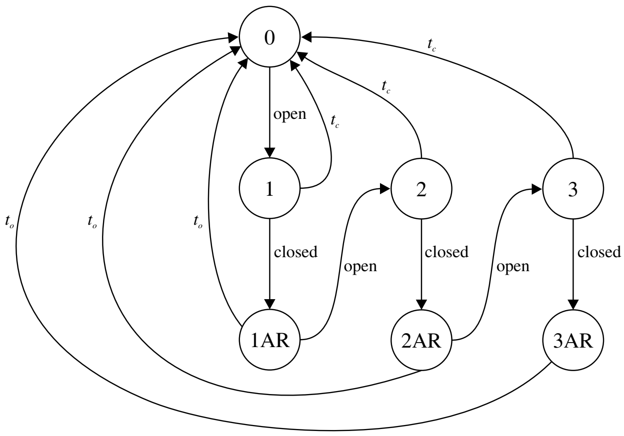

Figure 4 – Automatic reclosing state machine.

Local mode state machine

The local mode state machine is the simplest one. It is composed by two states: remote and local, as shown in Figure 5. The state change is given by respective triggers.

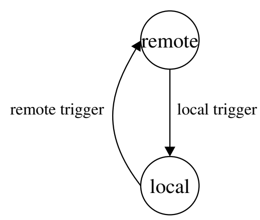

Figure 5 – Local mode state machine.

Faulty transformer state machine

The state machine for a faulty transformer is also very simple: normal and fault states as shown in Figure 6. The state machine enters the fault state after its respective relay 86 actuation. Every event in the same substation within a time t_f (typically 1 second) may be associated to the faulty transformer as an approximation. Therefore, after elapsed this time, the state machine comes back to its starting state, even if the transformer is still in fault.

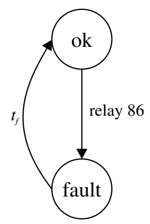

Figure 6 – Faulty transformer state machine.

Global state machine

The three previously presented state machines run concurrently in a single global state machine. Hence, priorities must be stablished among them. A natural priority for delivering alarms to a specific state machine is, from higher to lower priority, local mode operation, transformer fault and automatic reclosing. This priority is considered even if a lower priority state machine is not at starting state. For instance, if circuit breaker is waiting for its second reclosing and its bay enters in local mode of operation, then the reclosing group is finished and a local mode group is started.

Alarm sorting

Alarms may arrive out of order in the alarm processor, so that they have to be reordered and reprocessed. This can be efficiently implemented by keeping the state at time $t - \Delta t$, so that the processed alarms at time t are given by sorting and reprocessing all alarms from $t - \Delta t$ to t every time a new alarm arrives at time t. In this algorithm, Δt is the maximum delay for an alarm to arrive in the alarm processor. Alarms that arrive later than Δt are simply neglected. The procedure of sorting an alarm is depicted in Figure 7.

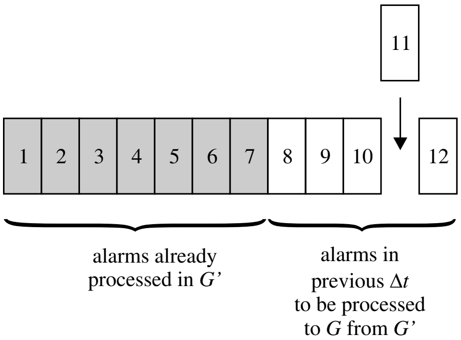

Figure 7 – Alarm sorting: a new alarm is sorted in the buffer and processed to generate groups *G* starting from already processed groups *G'*.

Alarm processing

After updating the global state machine of the alarm processor, the alarm is possibly assigned to a group of alarms. This assignment is straightforward considering the time, location and type of the alarm given the current state of the global state machine. For instance, every alarm generated in a substation where a transformer is in fault state, is grouped altogether. With proper data structures to compare location and time between new alarms and active groups (e.g. typically hash tables), this alarm processor may achieve really high processing rates, even for large amount of substations and devices.

The time sorting buffer is kept outside the alarm processor itself: the buffer is updated and reprocessed every time a new alarm comes in. It would be faster to treat sorting in the stating machine, however much more complex to implement and keep the code.

PRACTICAL PERFORMANCE

A prototype alarm processor coded in MATLAB using an implementation of the state machine described in this work was able to process 1025 alarms per second in a notebook for a real CEMIG's log data with 36.182 alarms from a single stormy day, as estimated in Figure 8. This is more than enough to provide a real-time response for the operator, considering the typical maximum density of 1,000 alarms per minute shown in Figure 9. In this study case, 55 groups were identified, all of them related to the automatic reclosing group.

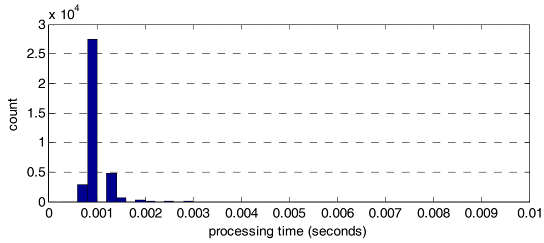

Figure 8 – Histogram of the average processing time of each alarm in a sliding window of size 50.

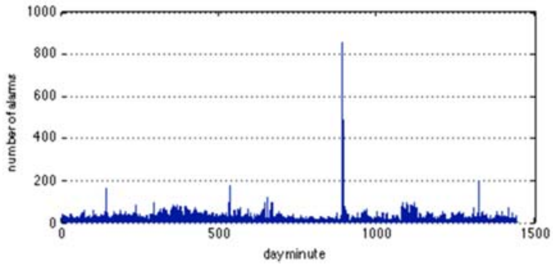

Figure 9 – Number of alarms in each minute in a typical stormy day.

Due to the granularity of the algorithm, the performance presented in this paper may be much improved if the state machine is implemented in a compiled language (e.g. C, C++).

CONCLUSIONS

The alarm processor presented in this paper is very specialized. It starts from a careful observation of the system, daily accumulated by operators, and ends in a complex state machine. All this work is rewarded by a fast alarm processor with an acceptable accuracy. Indeed, in the case study present in this paper, the results were 100% correct relatively to what is expected from it. This is a typical trade off in engineering: generality and speed.

A hybrid approach may be considered, since different approaches have distinct advantages. For instance, for real time, a strategy like in this paper may be suitable. For a post event analysis, an exact approach is more suitable. Search for patterns may help identifying groups of alarms to make future specialization into state machines.

Acknowledgments

This work was support by a P&D ANEEL project by CEMIG D, Brazil.

REFERENCES

[1] A. Bauer, A. Botea, A. Grastien, P. Haslum, J. Rintanen, 2011, "Alarm processing with model-based diagnosis of event discrete systems", Proceedings of the AI for an Intelligent Planet.

[2] W. Guo, F. Wen, Z. Liao, L. Wei, J. Xin, 2010, "An analytic model-based approach for power system alarm processing employing temporal constraint network", *IEEE Transactions on Power Delivery* vol. 25, nº 4, 2435-2447.

[3] P. C. Fritzen, J. M. Zauk, G. Cardoso Jr., A. L. Oliveira, O. C. B. Araújo, 2012, "Hybrid system based on constructive heuristic and integer programming for the solution of problems of fault section estimation and alarm processing in power systems", *Electric Power Systems Research* vol. 90, 55-66.

[4] F. B. Leão, R. A. F. Pereira, J. R. S. Mantovani, 2010, "Fault section estimation in electric power systems using an optimization immune algorithm" *Electric Power Systems Research* vol. 80, 1341-1352.

[5] J. Meléndez, O. Quiroga, S. Herraiz, 2012, "Analysis of sequences of events for the characterisation of faults in power systems", *Electric Power Systems Research* vol. 87, 2-30.

[6] L. Wei, W. Guo, F. Wen, G. Ledwich, Z. Liao, J. Xin, 2011, "An online intelligent alarm-processing system for digital substations", *IEEE Transactions on Power Delivery* vol. 26, nº 3, 1615-164.

[7] Z. A. Vale, A. M. Moura "An expert system with temporal reasoning for alarm processing in power system control centers" (IEEE Transactions on Power Systems volume 8 number 3 August 1993 pages 1307-1314)

[8] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, 2004, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", *IEEE Transactions on Knowledge and Data Engineering* vol. 16, nº 11, 1424-1440.