

Fault diagnostics and reporting in mobile services

João Paulo Pinto Trindade

Instituto Superior Técnico,
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal

Abstract. With the increase of enterprise network resources, performing management functions has become an extremely complex process. The widespread of different equipment and the non standardization of network event reporting has originated the creation of large repositories of data that don not follow any predefined structure.

This article discusses the problems of developing a platform capable of launching management actions based on the analysis of unstructured and heterogeneous network events. It then presents a possible architecture constituted by various distinct nodes divided into layers, each one implementing a different functionality, that have the objective of solving the enunciated problems.

Keywords

Event Reporting, Management Platform, Event Standardization

Introduction

Management platforms play an increasingly important role on today's organization [1]. Unfortunately building such solutions isn't trivial as the various critics on existing systems prove [2] [3].

With the spread of new technologies, an enterprise possessing information technology equipment can create and store an enormous quantity of raw data, which in many cases, is acquired without following any rigid structure defined in an information architecture. The result of this behaviour is a total chaos in log files and a near zero capability of managing the events reported by this data. It is with this premisses taken into account that originates the necessity of creating a platform able to launch management actions, based on the analysis of unstructured information.

This article focus its research in a concrete situation. Tecmic is a corporation established in Portugal that provides a fleet management service to its clients. The communications of the various vehicles that are geographical distributed with the central server originates a large number of events that are stored in several log files. Unfortunately, some of these events aren't conveniently structured. They range from a verbose textual structure to a hexadecimal representation of the communications that occur between the central server and the various vehicles.

The goal of this study is to provide the techniques and methodologies that allow the automatic launching of management functions and the creation of business intelligence reports, assuming that the structure of the events that originated such functions is only known when configuring the solution.

Related Work

There are two main topics of related work. The first topic will focus on how data retrieval algorithms can allow the discovery of a structure in unstructured events and how defined log formats can record event information. This study on data is needed because of the nature of the unstructured events intended to be structured. It is necessary to know how this data can be recorded and how to retrieve a structure information from a repository of unstructured events.

The second part of the research will incite on how it is possible to define management rules and investigate how the delegation of managing responsibilities can be achieved.

Data Retrieval

Data mining, also referred as knowledge discovery in databases, means a process of extracting implicit and previously unknown but potentially useful information from raw data in terms of knowledge rules, constraints, regularities [4]. There are many examples of organizations that have large amounts of operational network data repositories, which typically contain hidden knowledge that is crucial to some of the key tasks involved in effectively managing a network management system [5]. The studied techniques, presented in table 1, allow the discovery of what information can be managed in the organization, based in the unstructured recorded information [6].

Table 1. Log Mining Method Comparison

Method	Ease of Use	Performance	Predefined Attributes
Statistical Analysis	Medium	Low	No
Simple Log File Tool	Very High	High	No
Sisyphus	Low	Medium	Yes

As it is possible to see, the easiness of use and high performance of the Simple Log File Tool makes it one adequate system to analyse large repositories of events present in log files. The use of the simple Statistical Analysis, do not scale well when the complexity grows, and the fact that Sisyphus require predefined attributes provides a serious limitation when the events do not contain any explicit form of structure.

Standard Log Formats

Logging is a fundamental requirement of any system, as things will go wrong, we need a way to diagnose and isolate the cause [7]. There are several possible well defined formats which can be used to structure today's log files. The studied formats are listed at table 2. It is possible to conclude that the more structured and scalable formats have a very limited popularity in today's event repositories. By opposition and mostly due to historic reasons, non-scalable formats like BSD Syslog and NCSA have a high level of acceptance.

Table 2. Comparison Log Formats

Log Format	Popularity	Predefined Attributes	Scalability	Space Overhead
BSD Syslog	Very High	Yes	Low	Low
ELFF	Draft	Yes	High	Low
ULM	Draft	Yes	High	Low
XLF	Draft	Only One	Very High	Very High
NCSA	High	Yes	Low	Low
ODBC	Low	No	High	Low

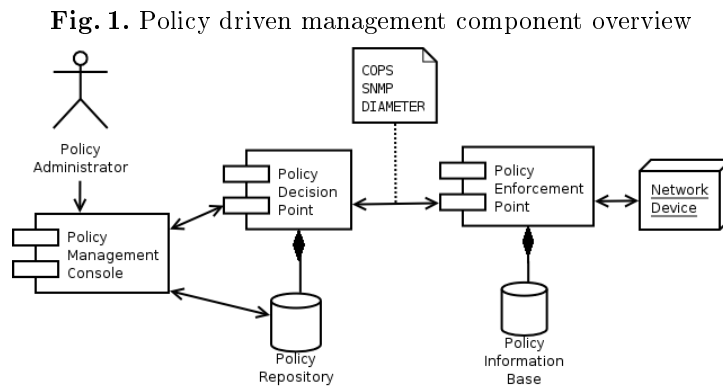
Policy Driven Network Management

Policies are central to network management to the point that industry has already started standardization efforts related to policy description and manipulation [8] in order to allow the abstraction of vendor-specific parameters [9]. Some work in formal models and methodologies to express policies can be seen in [10] [11] [12].

Policies can be formulated as sets of low-level rules that describe how to configure a device and how to manipulate the different network elements under different conditions [13]. In this way, a policy describes principles or strategies for a plan of action designed to achieve a particular set of goals identified by the managers of the systems [14].

Since configurations details are hidden from the policy administrator, he only needs to focus on defining the policies, independently of the different types of network resources. Policies provide a way to consistently manage multiple devices deploying complex technologies.

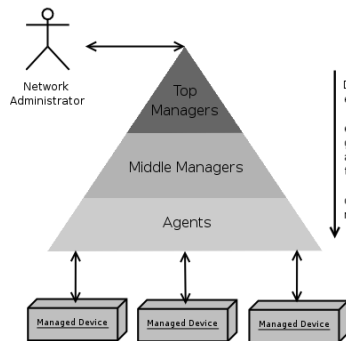
The basic components of of the policy framework defined by the Internet Engineering Task Force and the Distributed Management Task Force can be seen in figure 1.



Proposed Solution

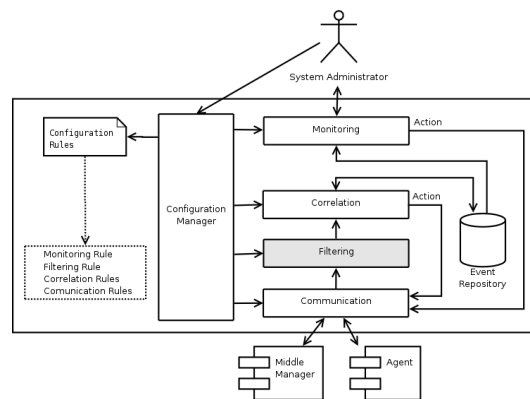
The proposed solution follows a tree like distributed paradigm and is constituted by three different components as shown in figure 2.

Fig. 2. Distributed Network Management System



The top manager system, shown in figure 3, presents an interface to the network administrator and communicates with both middle managers and agents.

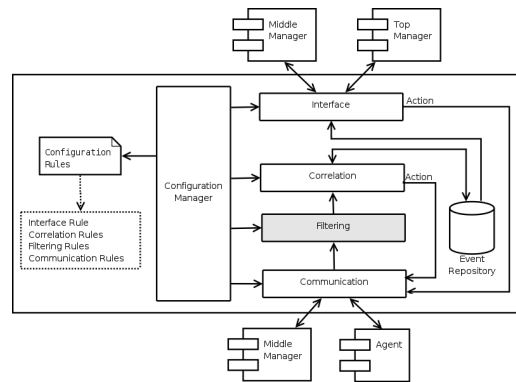
Fig. 3. Top Manager System



The middle manager, presented in figure 4 acts like a top manager but without the monitoring layer. Instead it has an interface layer that exports events recorded in agents or other

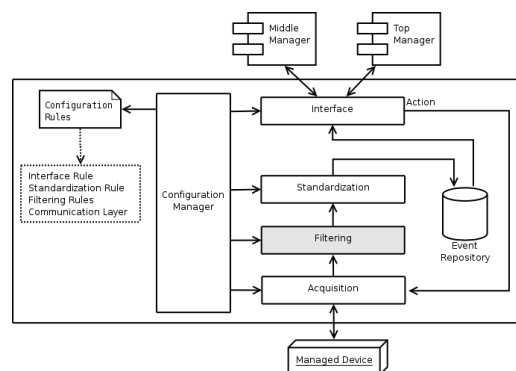
managers and provides the reception of delegation commands. The data exported in this layer follows a simple vector approach like SNMP.

Fig. 4. Middle Manager System



In the agent shown at figure 5 the communication layer serves the purpose of collecting data and issue actions to the managed devices. Its implementation will change according to the specificities of the device it communicates. The standardization layer is responsible for structuring the events in a way that is recognized by all other entities that constitute the network management system. Another property of the agent is that its event repository can be much simpler than those implemented in the managers. Besides only have to store events from one source, its size can be considerably smaller since managers will frequently fetch information from the agent.

Fig. 5. Agent System

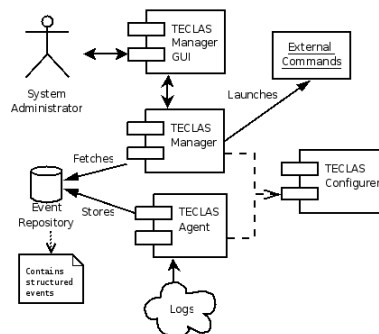


Implementation and Test

The implementation of the previously presented architecture is achieved by the TECLAS prototype which is a recursive acronym for: TECLAS is an Extremely Configurable Log Analyser System.

Graphically, the five main components that constitute the proposed network management prototype are presented in figure 6.

Fig. 6. TECLAS Main Components



The TECLAS Agent is the module responsible for parsing one or more unstructured log files and, through the combination of chained regular expressions, create structured events which are stored in the structured event database. A structured event is an object composed by one or more event values. Each event value has three fields, its name, value and type.

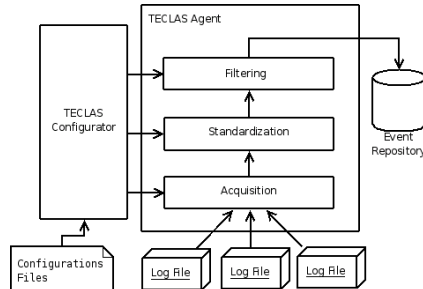
As figure 7 demonstrates, TECLAS agent module is constituted by three layers which have their behaviour defined accordingly with the configuration provided by the TECLAS Configurer.

The lower layer, acquisition, fetches unstructured events from the log files. It keeps track of what was the last line acquired and returns only the new events that show up in the log file. It also supports log files in which the unstructured events are contained through multiple lines. In these cases the configuration must provide the separator, or list of separator characters, that distinguish different unstructured events.

In the middle, the standardization layer, through the use of regular expressions transforms unstructured events into structured events. Each unstructured event is parsed by a list of transformations associated with the log file from which it originated. Each transformation only produces a value name field that depending on the transformation type can be an integer, a string, a priority or timestamp.

Several regular expression are contained in each transformation. The TECLAS Agent provides four native type of regular expressions: Contain, delete, match and replace, that can be sequentially chained to improve their functionality, each one returning the text that serves as an input to the next one.

Fig. 7. TECLAS Agent Module

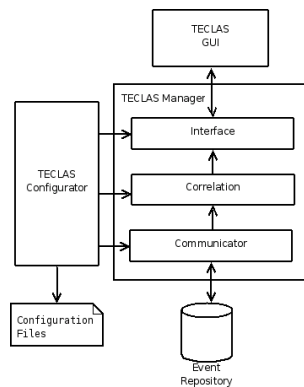


At the top of the agent, the filtering layer is responsible for deciding which structured events are valid to be stored. It can reject some structured events based in the missing of some event value fields.

The TECLAS Manager is the component of the TECLAS prototype responsible for correlating the events structured by the various TECLAS Agents and launching actions according to the configured correlation rules.

As figure 8 indicates, TECLAS Manager is constituted by three different layers that provide its functionalities.

Fig. 8. TECLAS Manager Module

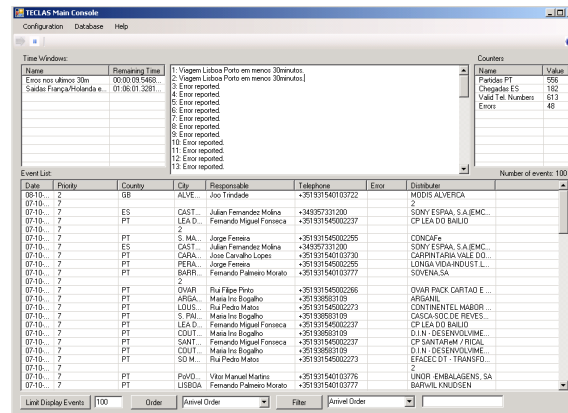


The TECLAS Configurer is the component that provides the others modules with the configuration that dictates their behaviour. Its function is to open a number of XML configuration

files, parse its content, and construct objects that represent the properties of those configurations.

The Graphical User Interface of the TECLAS System is provided by the TECLAS GUI component. There are two main type of windows: The main console and the new/edit configuration forms. The main console, is the primary form of communication between the network administrator and the TECLAS prototype. It provides a visualization of the various structured events that are analysed by the TECLAS Manager, a table containing the various time windows, another one containing a global counter list and a text box urgent messages are presented to the network administrator. In the new and edit forms it is possible to graphically edit or create agent and manager configurations, than can then be saved to a XML configuration file.

Fig. 9. TECLAS GUI Main Console

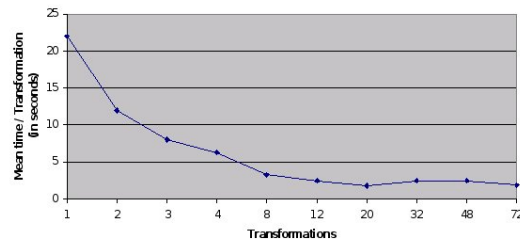


The TECLAS prototype was subject to the several functional and performance tests but only these last ones will be listed. For their execution a small command line program, the TECLAS Event Generator, that simply reads a list of events from a file and that accordingly to the specified time cadence interval, writes events in another file to be read by a TECLAS Agent was created. This program is intentionally simple which allows it to have a small footprint in terms of memory and central processing unit cycles. All test cases were executed in a fresh install of the Windows XP SP2 operating system. It was used the .NET Framework version 2.0 and the Microsoft SQL Server 2005. In terms of hardware specifications, the computer used had a Intel Pentium 4 1.3GHz with 512MB of random access memory. To measure the time taken to perform the defined tasks, the code of the TECLAS prototype was temporary patched in certain locations adding this way the option to print system time.

The test case number 1 measures how much time the TECLAS Prototype needs to structure unstructured events depending on the number of transformation rules. It is possible to see from figure 10 that as the number of transformation rules increases, the mean time spent

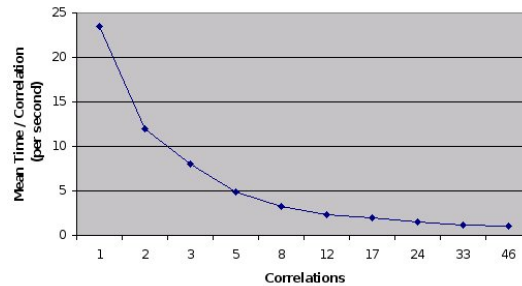
by the platform to process each unstructured event decreases in an approximately logarithmic order.

Fig. 10. Mean time by transformation



The test case number 2 measures how much time the TECLAS Prototype needs to correlate structured events and launch the respective actions depending on the number of correlation rules. It is possible to see from figure 11 that as the number of correlation rules increases, the mean time spent by the platform to process each unstructured event decreases in an approximately logarithmic order.

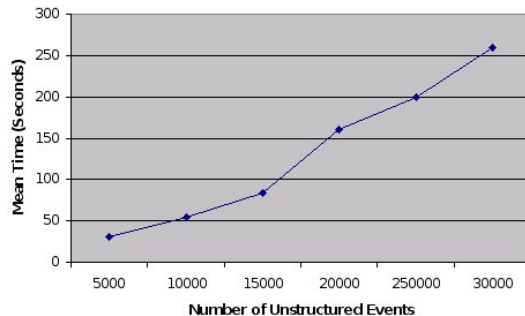
Fig. 11. Mean time by correlation



The test case number 3 measures how much time the global TECLAS Prototype needs to structure unstructured events, correlate structured events and launch actions depending on the number of unstructured events. In figure 12 it is possible to see that as the number of unstructured events increases, the mean time spent by the platform to process each one of this events increases linearly.

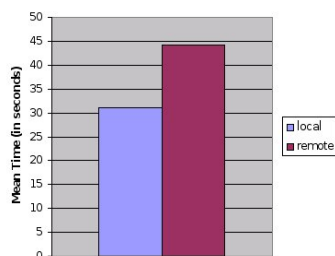
The test case number 4 measures how much time the TECLAS Prototype needs to structure unstructured events, correlate structured events and launch actions depending if the

Fig. 12. TECLAS prototype execution time varying the number of events



database management system of the agent is in a remote location or installed locally. The graphical representation of the results can be seen in figure 13.

Fig. 13. TECLAS prototype execution time varying the agent database location



Conclusions

This article has described the formulation, design, and implementation of a platform capable of launching management functions based on unstructured events.

We've started by studying tools that allow an organization to discover important information in their repositories therefore knowing what can be managed. After that, six different log formats were compared with the intention of understanding how its possible to describe events.

This paper looked at some new concepts that try to overcome the deficiencies associated with today's network management platforms. It was possible to see that the new trend is to allow the network administrator abstract from the most technical details of the various managed devices, and focus more on the service levels and business metrics of the organization that operates the network.

The proposed architecture presented a distributed platform where the system was divided into three elements each one with specific roles: the agent, the middle manager and the top manager. Each one of this elements was divided into several layers, which is a concept that defines a group of classes that share a common functionality.

The implementation of the TECLAS prototype allowed to demonstrate how the abstract design previously presented could be mapped into a concrete application. The tests performed allowed us to conclude it can be an useful tool for real production environments. The tests demonstrated that the execution time scales linearly with the increase in the number of unstructured events present in a log file.

Future Work

To finalize the article, this section will present the projects directions for future research.

The integration of management policies with the proposed architecture isn't still defined. This functionality, as been studied and its incorporation would prove to be a valuable benefit to the TECLAS system.

Another proposed future work is the addition of tools that allow the discovery of a structure in network events. Some of these tools were analysed in the state of the art chapter but their adaptation the proposed architecture was not done.

Finally, the TECLAS system is only a prototype and is not ready for a production environment. Future work in developing this system is necessary.

References

1. Goldszmidt, G.S., Schönwälder, J.: Integrated network management vii, managing it all, ifip/iecc eighth international symposium on integrated network management (im 2003), march 24-28, 2003, colorado springs, usa. In Goldszmidt, G.S., Schönwälder, J., eds.: Integrated Network Management. Volume 246 of IFIP Conference Proceedings., Kluwer (2003)
2. Pras, A.: Network management architecture (1995)
3. Cheikhrouhou, M.M., Conti, P., Labetoulle, J.: Intelligent agents in network management, a state-of-the-art. *Networking and Information Systems* **1**(1) (1998) 9–38
4. Zheng, Q., Xu, K., Lv, W., Ma, S.: Intelligent search of correlated alarms from database containing noise data (2002)
5. Garofalakis, M., Rastogi, R.: Data mining meets network management: The nemesis project (2001)
6. Burns, L., Hellerstein, J.L., Ma, S., Perng, C.S., Rabenhorst, D.A., Taylor, D.: A systematic approach to discovering correlation for event management. *Proceedings of the 7th IEEE/IFIP International Symposium on Integrated Network Operations and Management* (2001)
7. Eaton, I.: The ins and outs of system logging using syslog. *GIAC Security Essentials Certification* (February 2003)
8. Bhatia, R., Lobo, J., Kohli, M.: Policy evaluation for network management. In: *INFOCOM* (3). (2000) 1107–1116
9. Verma, D.C.: Simplifying network administration using policy-based management. *IEEE Network* (March/April 2002) 20–26
10. Wies, R.: Policies in network and systems management - formal definition and architecture (1994)
11. Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., Waldbusser, S.: Terminology for Policy-Based Management. *RFC 3198 (Informational)* (November 2001)
12. Yavatkar, R., Pendarakis, D., Guerin, R.: A Framework for Policy-based Admission Control. *RFC 2753 (Informational)* (January 2000)
13. Hasan, M.Z.: An active temporal model for network management databases. In: *Integrated Network Management*. (1995) 524–535
14. Lobo, J., Bhatia, R., Naqvi, S.A.: A policy description language. In: *AAAI/IAAI*. (1999) 291–298