

Proposal of application of Symbolic Learning and Data Mining for Domains in Event-Oriented Supervision Systems (in Osmius)

In the growing world of interconnectivity of users and applications, service orientation and contract agreements with users and clients about performance and availability of the offered / hired services gain more and more meaning. To diagnose before and better the problems presented in the systems that support these services can differentiate some systems and suppliers from others. It is intended in this article the use and application of Artificial Intelligence for the attainment of clear objectives in a field in which can be summed up with measurable and beneficial results.

The main proposals that are exposed in this article are: the division by abstraction domains in the algorithms of events correlation like in the implied actors, and the use of open source in a platform that applies many of the paradigms from the Object Oriented programming (Osmius), like the concept of data flows or "Streams" in the pattern "Pipes and Filters" and the application of Data Mining to the events databases. Besides, ambitious future lines are proposed in the environment of prediction of events or of resources capacities of the supervised entities.

Author:

José Luis Marina

Translated from Spanish to English by:

Samer Hasan Collado

Creative Common Work

Content

Introduction.....	2
Systems Supervision.....	3
Events Correlation and Automatic Learning	7
Introduction.....	7
Current Situation.....	7
Data Mining and Events.....	10
Proposal.....	10
Future.....	19
Conclusions.....	20
References.....	21

Introduction

The applications that are object of this article, are based on the paradigm of events based communication or, as well as it is known, publication / subscription of events notification. The range of applications is extensive and goes from desk applications, though real-time software, traffic control (including air and rail traffic), till the management of stocks transactions and electronic commerce.

The acceptance of this paradigm is broadly extended as we can check in its incorporation to standards like CORBA (Common Object Request Broker Architecture), SOA (Service Oriented Architecture) and JMS (API of messaging services of Java), and to commercial systems as TIBCO. A very important domain of events based applications is Systems and Networks Monitorization. Inside this domain we can find applications oriented to the supervision of systems of communications, services and computer systems, and those more oriented to the industrial world as the systems SCADA (Supervisory Control and Dates Acquisition).

As example of practical application we will use Osmius, a software for the gathering of events of distributed systems in network used for the systems supervision. Osmius is software of open source and it is based on open platforms based on frameworks and design patterns. In short it uses the work environment ADAPTIVE Communication Environment [ACE] that allows the re-utilization of code for multiple platforms with an excellent performance even in real-time, besides being open source very used in the international academic and market community and fruitful in research articles.

Initially the supervision and monitorization were oriented to the specific devices that composed our network. In the last years an orientation change is observed toward the Services supplied by the monitored system and that depend on the devices. The revolution that has meant Internet and the necessary high connectivity levels for the constant communication between users and companies, has hardened the performance and availability demands of many suppliers that are expressed by contracts of Service Level Agreements (SLA) with penalizations in case of non fulfilment. In this environment it is fundamental to have monitorization systems prepared for the reactive and proactive maintenance of the systems and devices and services.

These systems are based mainly in their interaction with an expert human, who assists the events and final alarms in a centralized global console or organized by domains. In occasions great amount of events are generated, that the administrator should filter to be able to identify the origin cause of the problem as soon as possible, to execute the necessary actions to re-establish the service or to solve its degradation.

It is in this context where the automatic learning deals perfectly: the improvement of performance of the identification process of real causes.

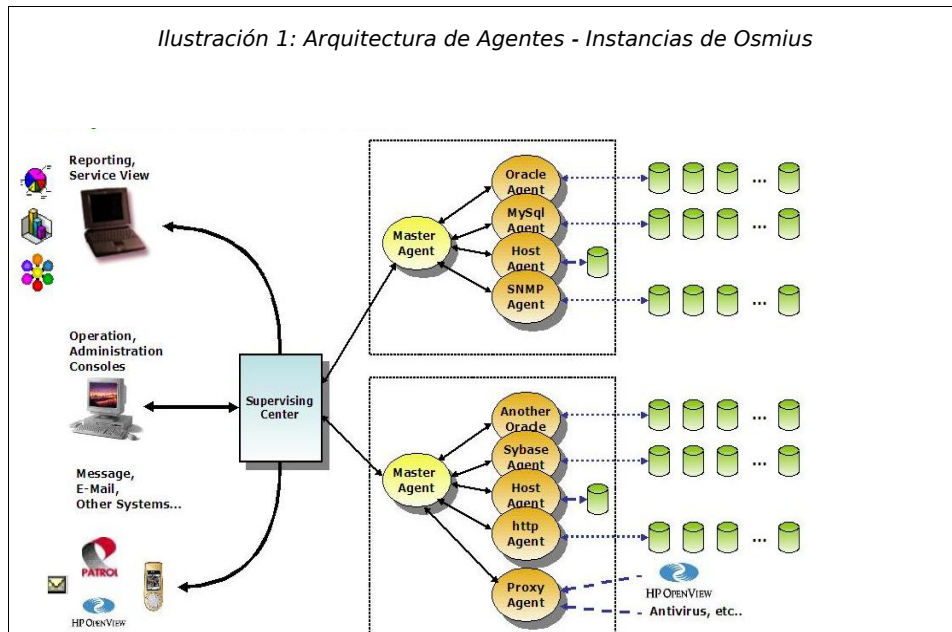
In the rest of the article we will explain on what it consists the events correlation and which ones are the techniques more used together with their principales advantages and disadvantages. We will see that exist available applications in the market with their inconveniences. We will explain the basic concepts inside the supervision of systems oriented to service basing on the software architecture of Osmius, and we will propose a system for the improvement of the process of identification of real causes and answer adapted in supervision systems. This system will be based on the distinction of experience domains and in the use of predicting algorithms and the data mining.

Systems Supervision

The supervision of systems is getting a great protagonism since the services that are lent now from the Data Processing Centers, (DPC) tends to be more demanding from the point of view of the availability, provoked overalls for the new services to the users of Internet like it can be the electronic-commerce.

In this applications domain we can speak about a wide application of **Manager-Agent Paradigm** in which agents are distributed in the infrastructure to monitor, and they take charge of collecting events and send them to a central manager for their presentation and/or processing. This architecture can present performance problems in environments with a big amount of elements and bottle necks can appear in the central server, as well as it is certain that a good design of the unfolding architecture and of the number and quality of information of the events to collect can palliate this problem.

Some groups [JPMARTIN2004_01] distribute the events correlation and their filters among the agents of the systems giving place to what they call **Self-managed Systems**. This way they let the systems the responsibility to filter what they don't consider important for the central supervisor. The agents would be more intelligent but more demanding and complicated, and more intrusive in the system to monitor.



Initially the supervision systems were very oriented to the world of communications and networks, and they had as main main character the devices that conformed the architecture. This type of systems was oriented to very technical users with a very specific formation to understand the presented information and to act in consequence.

Nowadays a monitorization system should be focused to service in opposition to those oriented to device.

Osmius proposes a modular architecture to be able to find a balance between both focuses keeping the best from each one.

For Osmius any possible origin of an event receives the name of **Instance**, doesn't matter if it is a final hardware device, a server, a database, a ftp service or a sensor of pressure or temperature. Each instance type has associated and agent's type, that is the one that knows how to ask for events to its instances and it is integrated perfectly in the rest of the architecture Osmius to send the events and to receive orders and indications on the events. The agents are integrated in the architecture through **Master Agents** that also take charge of receiving and sending to the **Server Central** the events and of receiving commands and to pass them to each one of the agents.

This way it is possible to leave independent the architecture of the monitorization system of the instances to monitor. The agents are just a bit intrusive and because of having a very defined interface it is very easy to develop and to insert new types of instances to monitor.

There is who distinguish among monitorization and events correlation oriented to device and oriented to service [AHANEMANN2004_01]. This article proposes a system able to abstract from this distinction to be able to reuse the learning and the correlation engine. Instance or Service are only two types of origins of the events.

Instance:

Origin of an event. Fundamental unit on which is built a system that we want to supervise.

Service:

Group of functionalities offered to a client or user by means of a conventional functionality. It can appear as a grouping of instances and applications.

Business Process:

Wikipedia: Group of logically related tasks carried out to achieve a defined result of business. In the context of monitorization supposes a grouping of services and availabilities necessary for this process.

Service Level Agreement (SLA):

Contract with the client or user regarding the performance and/or availability of an application, service or business process.

In Osmius the whole logical organization resides in the **data model** of the Central Server what allows us that the most technical orientation - Device or Instance - or more User oriented - Service or Business Process - is a presentation question according to the connected user's profile.

In the Table-1 the type of view is shown according to user's type. The interests about the information to show are not the same ones in the case of a network administrator interested in a specific model parameter that in the case of a directive that only wants to know if the billing process works like it is due or not.

User Profile	Element of Interest
Network Administrator	Instance: Interface, Router, Server...
Database Administrator	Instance: BD production, BD development,...
Application User	Application: Intranet (BD + Web Server +...)
Client	Service: Application or group of Applications
Executive	Business Process: Group of Services

Osmius decides to use an architecture Manager-Agent with light agents, and with logical groupings for profile to fulfill the expectations of technical orientation for the system users and of business orientation for the clients or business leaders.

User's type also conditions the perspective type before the appearance of a problem.

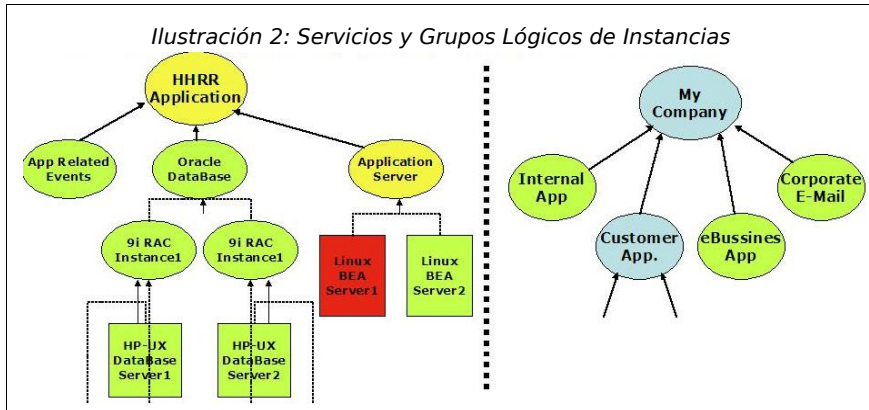
- Top-Down : (From top to bottom)

A service fails. Why is it failing? Which is the element that is causing the fall or degradation?

- Bottom-Up : (Of below to up)

An instance fails. What services are affected and in how measure? Which clients / users suffer the consequences?

■



If we add the contracts that SLA supposes with the users of the object systems, it appears all its sense the use of systems of agile monitorization, able to show different perspectives of the model and that are able to reduce the time of identification of real causes to problems being helped by the events correlation and of appropriate technics of automatic learning.

Types of Devices or Instances
Network Element: Router, Firewall, Switch,...
Server: Linux, Windows, Solaris,...
Database: Oracle, MySQL, Sybase,...
Internet Service: http, ftp, DNS,...
Application: CRM, Serv. Web...
Industry: Temperature Sensor, P Valve
Domotics: Switches, Appliances.
Others.....

Events Correlation and Automatic Learning

Introduction

The events correlation is an automated process that allows to an operator, administrator or user of an event oriented system, to find among many events those that are really critical in the environment.

It should be an automated process due to the great amount of information that can be presented. The events correlation allows to reduce big quantities of events to more reduced groups and with more meaning on its cause and – therefore - its possible solution.

This article proposes to use the same correlation techniques for events oriented to device and events oriented to service, opposed to what was proposed by other authors. It is sought to apply the same correlation mechanisms to the different domains formed by groupings with every time greater level of abstraction.

This way each expert or user can validate the correlation rules of his specialization field and we obtain a lineal scalability of the performance, besides the obvious reusability of data mining and learning algorithms. Same algorithms; different domains. It continues being important that the correlation algorithms maintain a low complexity (OR (n) or OR (log (n))), due to the great number of rules that we can manage and to have to apply to huge amounts of events.

It has a critical importance in the current world of interconnected systems, to achieve the objectives of intelligent correlation so it helps to detect sooner and better the problems arisen in the supervised network. Doesn't matter the nature of the system, and every time with higher demands of availability and performance.

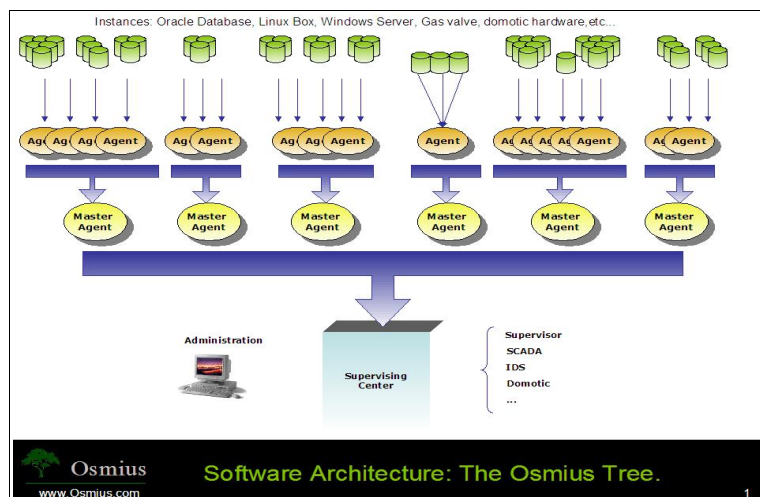
Current Situation

Different correlation strategies exist in function of the **place inside the architecture** in which is carried out this function.

- Central. In the central server to which all the events arrive.
- Agents Local.
- In the own network element.

Taking a more centralized or distributed strategy has advantages and inconveniences, and in this article it is proposed a mixed approach, configurable in execution time.

Strategy	Advantages	Disadvantage
Centralized	Controlled impact in the monitored services. Agents are simpler.	Excess of load in central server.
Distributed	Lower flow of events toward the central server.	Agents more complicated. Impact in the performance of the monitored element.
Self-management	A system distributed to the maximum.	Low abstraction of the final element and low reusability. Impact in the element. Low transparency.



The techniques used to carry out the events correlation are several:

- Finite State Machines.
- Rule Based Reasoning.
- Case Based Reasoning.
- Model Based Reasoning.
- Bayesian Networks
- Neural Networks

We can speak about two big groups of approaches to the correlation problem.

On one hand, those that need information about the complete system architecture, in the form of a components diagram and its relationships that then can be translated to a correlation matrix. This diagram can be directly given to the correlation system, or learning techniques can be used for inferring it from

the gathered data.

On the other hand, the systems that don't need previous information about the components and their relationships. They are usually based on rules in the way "IF this happens THEN do that". They receive the events and apply those rules, although they can also learn from the data to infer new rules.

Let's introduce some of the most used ones.

Model Based Reasoning (MBR):

Each component (instance, according to the Osmius nomenclature) is modelled regarding its attributes, behaviour and relationship with other models and components.

The correlation arises as the result of the collaboration among models.

Rule Based Reasoning (RBR):

A group of rules is used that are applied to the received events - usually in a certain time window - to correlate the events.

The algorithm of RETE, designed by Charles L. Forgy from the University of Carnegie Mellon, is one of the most efficient in searches of patterns to implement expert systems based on rules.

The groups of rules to apply can end up being huge and this makes difficult the maintenance of the system.

Also, if something not foreseen by the given group of rules happens, the system fails in its made. Many systems RBR are based in the recognition of regular expressions [RVAARANDI2002_01].

Code Book Approach

A Code Book is a recipes book where it is detailed step-by-step how to follow a procedure.

The same as RBR, it is based on a correlation algorithm. It uses as input a graph of dependences with events and root causes as nodes, and directed arrows that interconnect them. Once built the graph, it can be optimized deleting redundant information, and it becomes a correlation matrix.

The matrix has as columns the root causes and as rows the events. In the simplest form each cell has a binary value (it correlates or it doesn't correlate).

It allows to manage situations with unknown combinations of events. RBR is more flexible.

Reasoning Based on Cases (Based Marry Reasoning. CBR):

It doesn't need previous knowledge about the infrastructure. It has a database of cases that happened before, together with the root causes associated.

We can apply learning on the cases helping to identify real causes.

Data Mining and Events

All the commented strategies can be complemented with techniques of **Data Mining**, to identify rules, dependence graphs or conclusions on the supervised system.

[GUPTA] uses algorithms for **Data-Mining** that use performance data to obtain dependences among components based on probability.

This way we could generate the dependences graph treating a dataset instead of hand-making it. With this graph root causes of concrete problems can be found.

[ENSEL] suggests to use neural networks to generate automatically graphics of dynamic dependence among servers, although there are not experimental data about the precision of these methods. Besides, this approach doesn't detect causation, only correlation.

Data Mining mixes statistical techniques and data handling that provides us a form of grouping the data of the received events, to find interesting combinations [JLHELLERSTEIN1999_01].

This way there can be identified patterns for generating correlation rules.

In the case of events is very important the time variable, since there should be kept in mind the sequence of the events more than their only occurrence.

Proposal

This article intends to apply mixed techniques of automatic learning and data mining over events of a systems supervision system, called Osmius.

Osmius implements many of the nowadays investigated paradigms in computer science, and being its code of free access and modification allows its intensive use in the Artificial Intelligence research and in other fields related to Technology and Information.

The field of applications of network systems supervision has a leader interest inside the growing current frame of service supplier - client and SLA. The quick identification of root problems has a critical importance, and Artificial Intelligence fits as methodology and in this field it has simple indicators and objectives to measure its results.

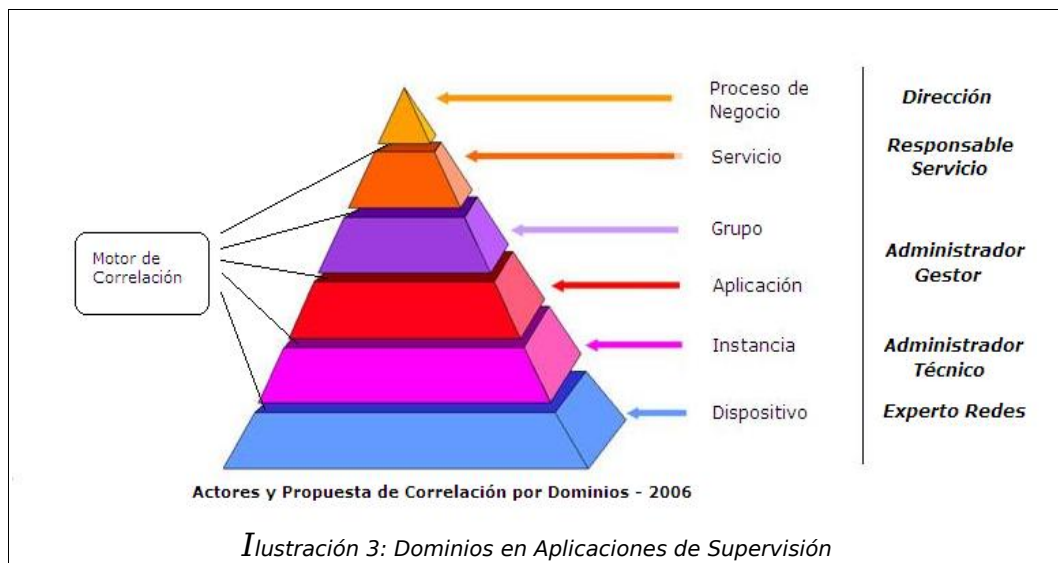
This field seems to promise results for Artificial Intelligence, because very mentioned problem of the lack of data, in this case, doesn't exist. It is easy to build a good events database of a monitorization system already implanted and to apply learning algorithms or techniques of data mining.

The main ideas to apply in the proposal are:

- Correlation Domains:

It is proposed to divide the events in correlation domains, in a way so the rules to apply and the user or system supervisor can change without influencing a group in the other one. It is different the client vision of a problem than the one of a network administrator. This division in domains

allows an approach and progressive reduction of events as it advances in abstraction capacity, and the reusing of algorithms.



- Opened Rule Based Engine:
- A basic and slight engine based on generic attributes of any events system. Standards, events structure, and open code and algorithms, will allow us to use the engine in several flavours: centralized, pure distributed, or in a mixed way, and depending of the type of supervised system, to adapt by parametrization.

The code and its documentation can be downloaded from Source-Forge, the web for supporting free software developments, in:

<http://osmius.sourceforge.net>

- Osmius - Productive domain for Researching:
Osmius will continue oriented to the practical and economic market of the supervision systems. Being "well made" and a completely modular system will allow to use directly the thesis and researches results, if they are well focused, towards a final use for certain users and/or markets. If these conditions are accomplished it can be the ideal environment for the implementation and observation in real environments of new thoughts and techniques, that in others contexts is so difficult and frustrating, regarding the lines researched by an university or research centre.

- Dynamic Correlation Rules. Methodology of Data Mining:
We propose the existence a priori of rules whose utility has already been proven in those monitorization systems, and the use of data mining together with learning algorithms, for presenting new rules to each one of the experts for its approval and setting on.
- Data Mining, automatic or attended, intends to be as main tool in a case, methodology and consultancy in the other one, to achieve as a result new

rules for feeding the correlation engine or reports and patterns to use for possible improvements like the management of resources of the supervised systems.

- To define the problem in the field of the automatic learning we need to define the problem. In this case it is fundamental the relationship of attributes that consists each event in Osmius. Its appropriate description and understanding help to outline the bases of the problem to solve by the algorithms of automatic learning and the techniques of Data-Mining.

Attribute	Description
TYP_MESSAGE	Type of the Message or Event. Normal - Internal - Error
COD_MASTER	Master Agent of which it comes. It informs about the architecture of the supervision system more than about the system monitorized.
COD_AGENT	Agent that originates the message. Idem that the previous field.
COD_MESSAGE	ID of the event type that originates the message (Percentage of Disk busy, seconds in responding to http petition, temperature of the core,...)
DAT_INIEVENT	Date and time with microseconds accuracy in which begins the question of the event.
DAT_FINEVENT	Date and time with microseconds accuracy in which it receives answer to the question of the event (we can identify questions or events that take "too much").
COD_INSTANCE	Unique ID of the instance or device from where the message comes (Name of the database, Serial number of the valve,...).
TYP_INSTANCE	Type of the Instance (A database, a valve, a toaster,...).
VALUE	Numeric Value of event (In Osmius every answer should be reduced to an integer number)

According the Master Agents are sending the messages (events) produced by their agents (monitoring the different instances), they will be introduced in the correlation chain. In this chain we have the different engines in a configuration based on the design pattern "Pipes and Filters." See *Illustration 4. Correlation Architecture*.

Each one of the processes is in fact a clone of another in such a way that only change the correlation rules to apply. Also, and in execution time, we will be able to activate or to stop each one of the engines being able to improve the adaptability of the different systems that can be supervised by the Osmius platform. If one of the processes finds correlation possibility in its input data, it applies it passing the redundant information to the historical of received events, and the result from the correlation to the following process in the chain or, if it is the last one, to the database of active events so that the operators or

administrators of the system manage its analysis and manual step, and if it 's OK, it will pass to the historical of events.

In the historical of events we can find the information point in which will be based the whole analysis problem and data mining with the objective of identifying occurrence patterns and generation of new rules for the different domains.

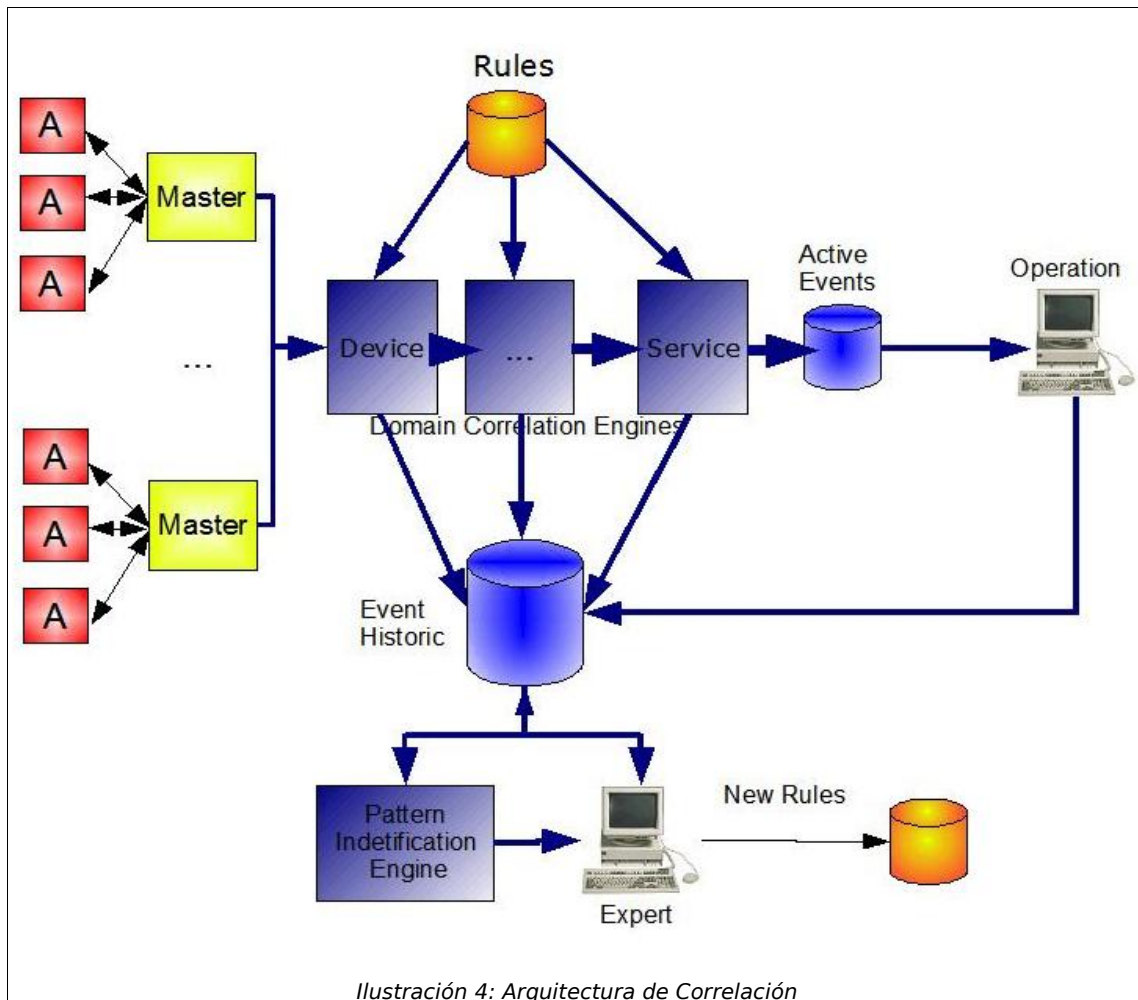


Ilustración 4: Arquitectura de Correlación

A engine will propose to the experts rules based on the found patterns that will organize the information to be able to reach to more conclusions and they will be those in charge of accepting new rules and to introduce them in the rules database, so that they can be applied by the domains correlation engines.

The cycle of life of a message or event is the following one:

- An agent with the ability of recovering data of a certain type of instances (TYP_INSTANCE) read from a configuration file.
- It recovers a instance name (COD_INSTANCE).
- For that instance reads a specific type of event or message to monitor (COD_MESSAGE).
- The agent begins the action to recover an associated value (VALUE) to the event or message type and an explanation text (TXT_VALUE) in a certain moment (DAT_FINEVENT).
- The agent builds a message with the appropriate fields and it sends them to the Master Agent.
- The Master Agent sends the message to the Supervision Centre.
- The Supervision Centre inserts the message in the table of **Active Messages**.
- The message will pass to the table of **Historical of Messages** through two main ways:
 - One of the correlation engines delete it from Active together with other messages or not, and it passes it to the Historical applying certain rules.
 - One of the users "recognize" the message and this way it is erased from the Active and falls in the Historical.

The messages, when they are passed to the historical, store the date and time of this step to be able to analyse the possible correlation among events according to their recognition date together with the date of event occurrence.

Inside the attributes associated to each message there are some that are not outstanding for the events correlation like the master agent's code that makes as *proxy* so that the message arrives appropriately.

The outstanding attributes of each message are then:

Attribute	Example / Format
Type of Instance	Oracle Database [ORA01], MySql Database [MYSQL], FTP Server [FTP01], Demotic Heating [SYM01], Industrial thermometer [TER01], etc...
Code of Instance	INST01, INST01,...
Code of Message	For an Oracle database:

Attribute	Example / Format
	Number of connected users [NUMUSU] Maximum Processes Percentage used [NUMPRC] Number of files of "redo log" that are lack to file [REDLOG] Percentage of occupation of the most used "Tablespace" [FRETBS] Ratio of success in the data cache buffer [BUFRAT] etc
It dates of the Event	20060707145959 / YYYYMMDDHHMISS
Value of the Event	Numeric
Text of the Event	This field is only descriptive and it will be useful in a detailed analysis of the received messages and it stops to check the utility of a correlation and to make more understandable the context in which the messages take place. "The Tablespace but busy it is [RBS1] con:82 percent. [90,95]"

In the data table assistant we can see a real dataset extracted at first hour of the labor day of an environment with more than 40 instances of databases Oracle giving productive services. This would be the type of data to analyze with the correlation engines and automatic appropriate learning algorithms.

In work to carry out consists on this case in:

- To determine periods of time in those that go up the values of certain message types for all the instances in general. For example the number of connected users at first hour of the morning or in the hours assigned for the lunch. It goes up the load of the database at nights when rushing the prosecution works for lots and the copies of security.
- To determine if for one it spoils instance there are tendencies of the values of different message type that corleación. For example when ascending or to lower the number of users also makes it the percentage of used processes.
- To determine its there are values of types of events or messages in different instances that they are only given together or they correlate strongly. For example whenever I lose the connection to TO (CONNECT = 1) I also lose it with B because they are in the same one server and network segment.
- To analyze if value high relative in a parameter during certain window of time provokes that other parameter also ascend or lower their value. For example if the percentage of processes used he/she stays high finally low the success rate in cache.

All these situations can be analysed to the margin of the experts in the instance types and observed parameters, and later on they should be interpreted in common to give him/her the technical final sense in case there was him.

DAT_FINEVENT	TYP_INSTANCE	COD_INSTANCE	COD_MESSAGE	VALUE	TXT_VALUE
20060626083835	ORACLE01	CAMBP	CONNECT1	0	Conexión correcta
			NUMUSU01	5	Cantidad de Usuarios Actual.
			NUMPRC01	10	El porcentaje de Procesos usados.
	GASBP	CAMBP	REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
			CONNECT1	0	Conexión correcta
			NUMUSU01	20	Cantidad de Usuarios Actual.
	HTBP	CAMBP	NUMPRC01	12	El porcentaje de Procesos usados.
			REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
			CONNECT1	0	Conexión correcta
	ATBD	CAMBP	NUMUSU01	25	Cantidad de Usuarios Actual.
			NUMPRC01	41	El porcentaje de Procesos usados.
			REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
20060626084439	ORACLE01	CAMBP	FREOBJ	0	Objetos sin capacidad para extenderse.
			BUFRACT	86	Porcentaje de Acierto Cache de Datos.
			CONNECT1	0	Conexión correcta
	GASBP	CAMBP	NUMUSU01	30	Cantidad de Usuarios Actual.
			NUMPRC01	44	El porcentaje de Procesos usados.
			CONNECT1	0	Conexión correcta
	HTBP	CAMBP	NUMUSU01	10	El porcentaje de Procesos usados.
			REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
			NUMLCK	0	Número de Bloqueos entre procesos.
	ATBD	CAMBP	CONNECT1	0	Conexión correcta
			NUMLCK	0	Número de Bloqueos entre procesos.
			NUMPRC01	13	El porcentaje de Procesos usados.
20060626090323	ORACLE01	CAMBP	REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
			FREOBJ	0	Objetos sin capacidad para extenderse.
			BUFRACT	86	Porcentaje de Acierto Cache de Datos.
	GASBP	CAMBP	CONNECT1	0	Conexión correcta
			NUMUSU01	29	Cantidad de Usuarios Actual.
			NUMPRC01	43	El porcentaje de Procesos usados.
	HTBP	CAMBP	CONNECT1	0	Conexión correcta
			NUMUSU01	5	Cantidad de Usuarios Actual.
			NUMPRC01	10	El porcentaje de Procesos usados.
	ATBD	CAMBP	REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
			CONNECT1	0	Conexión correcta
			NUMUSU01	19	Cantidad de Usuarios Actual.
GASBP	CAMBP	NUMPRC01	12	El porcentaje de Procesos usados.	
		REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]	
		CONNECT1	0	Conexión correcta	
HTBP	CAMBP	NUMUSU01	23	Cantidad de Usuarios Actual.	
		NUMPRC01	39	El porcentaje de Procesos usados.	
		REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]	
ATBD	CAMBP	FREOBJ	0	Objetos sin capacidad para extenderse.	
		BUFRACT	86	Porcentaje de Acierto Cache de Datos.	
		CONNECT1	0	Conexión correcta	
20060626094133	ORACLE01	CAMBP	NUMUSU01	35	Cantidad de Usuarios Actual.
			NUMPRC01	49	El porcentaje de Procesos usados.
			CONNECT1	0	Conexión correcta
	GASBP	CAMBP	NUMUSU01	5	Cantidad de Usuarios Actual.
			NUMPRC01	10	El porcentaje de Procesos usados.
			REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
	HTBP	CAMBP	CONNECT1	0	Conexión correcta
			NUMUSU01	24	Cantidad de Usuarios Actual.
			NUMPRC01	14	El porcentaje de Procesos usados.
	ATBD	CAMBP	REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]
			CONNECT1	0	Conexión correcta
			NUMUSU01	25	Cantidad de Usuarios Actual.
GASBP	CAMBP	NUMPRC01	41	El porcentaje de Procesos usados.	
		REDLOG	1	Ficheros de Redo que faltan por Archivar:[1] [7,10]	
		FREOBJ	0	Objetos sin capacidad para extenderse.	
HTBP	CAMBP	BUFRACT	86	Porcentaje de Acierto Cache de Datos.	
		CONNECT1	0	Conexión correcta	
		NUMUSU01	33	Cantidad de Usuarios Actual.	
ATBD	CAMBP	NUMPRC01	47	El porcentaje de Procesos usados.	

It is the data mining applied to the generation of rules by domain, together with the use of patterns of design of the architecture software, the one that you/they endow to the pattern of a great adaptability and flexibility to achieve the objective of improving the identification of problems and the answer to solve them.

Data Mining in the context of the Databases of Events

The data mining intends initially for its use on the data of the shopping cart of the purchase in supermarkets. In the Data Mining objective it was to respond to questions of the type: The purchase occurrence **A** indicates that **B** also buys? (Association Rule)

In this case the algorithm a priori proposed by Agrawal it provides good results in this type of situations and overall can give us the measure, being based on parameters called support and trust of a rules, of the good or strong that is between two the association events.

[Hellerstein] it already advances results of applying the data mining to monitorization systems based on events, in this case of systems of communications. These results show patron, experts as behaviour forms and grouping of the data, observed in most of systems.

We can explain the similarity between the problem of "the basket of the purchase" and the groups of events, containing these inside a window of time and formed this way the baskets to analyze. It is very important also to select the attributes for those that to contain the different events.

Some of the patterns that repeat:

- **Pattern of storm of events:**

You he/she takes place when it fails something important for a part or all the one group of the system, and it doesn't have the one wanted (in principle) redundancy. A clear example is the failure of the server of names in a network: None of the questions of the servers on them same or others of their network will be able to be responded and they are generated a great quantity of failure events in the resolution of name and addresses. A pattern can also be continued in cascade in the one that the failure of an element goes spreading to other elements to the style of a reaction in chain that it provokes the storm of events.

By means of Data Mining can look for:

Periods in those that there are more storms than a certain threshold (let us remember the baskets)

To study thoroughly and to recognize patterns during those periods.

- **Rhythms:**

Occurrences repeated of the same event or group of events. According to [Hellerstein], this type of repetitions is very common in those systems of network supervision and they are given in 50% or 60% of the events. Said in another way, until 60% of the events they can to associate with a repetitive pattern with certain rhythm. Interesting and logical, if the events

of for yes they are monitoring in fixed periods, like it is usually the one case.

In this case can enter so much in game the association algorithms of rules that only keep in mind the dependence with regarding the frequency (in this case of appearance), as other models that also keep in mind the variable time.

- **Dependences mutual:**

Groups of events that spread to happen together instead of making it in way separate and independent.

With this it is patent that the data mining, with more or smaller grade of attendance. expert cosultoría versus automatic algorithms of recognition of patterns. it allows to arrive to results as for achieving objectives of improving the performance of the supervision systems.

These automatic algorithms will be oriented in Osmius to the generation of rules that I lower "expert" supervision in each domain, they will incorporate or not to the group of rules initials and previous, of the engine of corresponding correlation. *To See Illustration 1.*

Future

Not it is room for doubt that the interest in systems able to supervise applications or business processes so that he/she improves the borrowed service it will continue in peak in next years.

As he/she goes maturing the installation of mixed systems and with patent results as he/she intends in this article, new challenges some will be approached of which it is about advancing it is this section.

In this area they will charge bigger protagonism and interest the **Algorithms Predictivos**. The proposal in this article is to apply this type of algorithms to detect problems in a service, and to even act corrigiéndolos that they affect to the contracts signed with the clients or users of the the systems.

These algorithms will be capable of reponder in way proactiva not only before problems if not to forecasts of performance capacity, or of any resource type.

Questions as:

- Which it will be the use of CPU or disk of this server during the one next month?
- And during next minute?
- I can to predict if this router will fail in the half next hour?

And in general of the type:

- Which will the evolution of variable X be in the resource AND?

, they will be objective very interesting of the supervision systems in those that he/she will have application the research it has more than enough automatic learning and data mining in event oriented systems

The new systems and algorithms will be able to use the data of each one of the

received event (variables) types of the different instances to carry out the predictions, and to use the possible correlations with other variables to vary the estimates.

We will use algorithms of Analysis of Temporary Series to carry out predictions to short or I release term. In this case they seem the association algorithms that keep in mind the variable time to fit like they can be the *Hidden Models of Markov*.

Not it is disheveled that as they leave programming and using algorithms for these learning types, specific design patterns appear in the automatic learning.

It will also be exciting the identification of **patterns of work** methodology and their later documentation that will arise of the experience of the analysis of the databases of events to respond to the mentioned questions, and that they will serve other experts in data mining. To this respect it will be able to document and to be modelled a **methodology** for the different phases of the installation of the supervision of a system whichever it is their nature. They would be the phases of definition of requirements so much technicians and as of business and SLA, the definition of the requirements of no-functional of so much quality of the service like of the monitorization, as well as the phases that define the behaviour of the system and their refinement regarding results, generation of new rules and presentation of reports to the different agents inside each domain. In the refinement phase it will charge sense a specific **methodology for the automatic** learning in the context of association of events and generation of rules.

Conclusions

We have seen that the systems that you/they help to supervise and to monitor other systems in network take a great importance to help at these last to fulfill a quality of more and more demanding service and that it is captured in contracts with the users and/or clients through the Agreements of Level of Service.

To improve the productivity of the systems and of the operation groups in charge of supervising them it is important, and although this productivity has improved it continues being problematic what you rule of correlation to write and to apply in each environment.

Also, to keep in mind that inside each system different visions exist, far from complicating the problem, he/she helps when facilitating their understanding, and therefore in the design of the possible solutions. Not we can be about equal it forms the necessities of information in the supervision of a system of a techniques very specialized user in the final devices that the client's vision worried by the performance of their electronic commerce transactions, let us put for case.

Distinguishing the different forms of seeing oneself problem, it will be able to opt for the advantages of each a, and we will let him to be the system the one that can define the strategy that better he/she adapts to their performance characteristics, number of events and necessity of information in the central recolector.

The commercial tools for the events correlation of those that one has news

before writing the article present certain problems like they are:

- The **high one configuration complexity**, setting in march and maintenance. This impedes the division in domains for abstraction capacity and to the access to users with useful knowledge on their domain but not excessively technicians.
- They are usually **platform clerks**. The engines are developed for only to be used with the format proprietor of the supervision tool. Also, as user, you only have the binary ones compiled for a reduced number of platforms.
- **Expensive**.

A good tool of Open Code able to correlate events in different formats and the sufficiently robust thing to work in productive demanding environments, covers an evident hole in the academic and commercial current market.

References

[DCSCHMIDT1993_01]

The ADAPTIVE Communication Environment An Object-Oriented Network Programming Toolkit for Developing Communication Software.

Douglas C. Schmidt - 1993

11th and 12th Sun user group conferences. California

[DCSCHMIDT1999_01]

An architectural overview of the ACE framework. To Study of he/she Marries Successful Cross-Platform Systems Software it Reuses

Douglas C. Schmidt - 1999

USENIX

[JPMARTIN2004_01]

Distributed Event Correlation and Self-Managed Systems

Jean-Philippe Martin-Flatin - 2004

Proc. Int WorkShop on Self - * Properties in Complex hf. Systems

[RVAARANDI2002_01]

Platform Independent Event Correlation Tool for Network Management.

Risto Vaarandi - 2002

2002 IEEE / IFIP Network Operations & Management Symposium

[AHANEMANN2004_01]

Assured Service Quality by Improved Fault Management. ServiceOriented Event Correlation.

Andreas Hanemann et to the one - 2004

ICSOC'04 New York, USES

[JLHELLERSTEIN_01]

Minig event dates for actionable patterns.
Joseph L. Hellerstein and S. Ma - 1999