



# NETWORK MONITORING SYSTEM

<sup>1</sup>MOGAL AMINA BAGUM, <sup>2</sup>D MURALI

<sup>1</sup>Research Scholar, <sup>2</sup>Associate Professor

<sup>1</sup>Computer Network,

<sup>1</sup>Quba College of Engineering and Technology, Nellore, Andhra Pradesh, India

**Abstract :** - This paper presents the modeling, design, implementation, and evaluation of a prototype Network Monitoring (NM) information system. First, a general overview of such management systems is provided highlighting the need for the visualization of network elements in order reliable network monitoring to be achieved. Moreover, the importance of monitoring the network operation from the network administrators and other relevant users is underlined by presenting the main reasons for using NM tools. Then, the functional and architectural design and implementation of the proposed network monitoring tool are presented. This tool uses state-of-the-art implementation standards and technologies, like the TMF608 Multi-Technology Network Management (MTNM), the NML-EML (Network Management Layer-Element Management Layer) interface and the JBoss enterprise application platform. A major feature of this prototype NM tool is the graphical representation of the telecommunication elements/managed objects over a panel, which contains a map and representations of possible active alarms over these elements/objects. The proposed NM tool supports real-time alarms and allows the users to acknowledge them. Next, an operational demonstration of how the prototype NM information system works is provided along with some discussion on evaluation results. Finally, conclusions and future work are given.

**Index Terms** – Network Monitoring, Simple Network Management Protocol, Management Information Base, Event Dissemination, Management Information Systems, Network Operations.

## I. INTRODUCTION

Network monitoring (NM) is a broad field which includes, among other activities, device monitoring, application management, planning, security, maintenance, service quality and troubleshooting. It is well-established that NM plays a major role in the effective operation of a network and, thus, it is a central part of a networking management system. All these NM activities must be coordinated and overseen by experienced network administrators. The network data which can be seen using monitoring systems are very important for the effectiveness of an experienced administrator. The administrators should, always, not only have a thorough knowledge of what is happening in their network, but also access to real-time and historical data, as well as to performance and status data of each network device

The network monitoring services are used by the network control services to correct problems or improve network performance. The improvement of network performance is associated with the quality of the offered network services and their availability, meeting and exceeding the users' demands.

In this paper, a model for network monitoring is described and a technical approach towards the development of such a system is presented in detail. Specifically, the proposed NM system is based on the graphic depiction of telecommunication network objects. Through appropriate programming, a variety of icons can be used for the representation of these objects, whereas each object contains a description. The application

interface is enhanced by providing a background map. The objects can be moved on the interface and their desired position can be saved, while they can be connected to each other with a link object. The objects accept SNMP traps (alerts), so that each one of them can act independently (NET-SNMP, 1992).

In this paper, we first present the importance of network monitoring and its basic principles. Then, we focus on the applied network management model and, specifically, on the object types the system can manage. A description of the system architecture and the way the alerts should be managed follows. An operational example of the system is also presented. Moreover, the usefulness of the system is stressed using a statistical performance example.

The paper is organized as follows. Section 2 provides the definition of network monitoring and a literature review that highlights which layers of the OSI (Open Systems Interconnection) model NM relates to through the presentation of related network monitoring systems. Section 3 outlines the problem space and underlines what should be monitored and why and it gives an example of the effects of the lack of an efficient NM information system in network operations. Section 4 analyses the proposed system and its components, thoroughly. Section 5 demonstrates how the application works with three added Network Elements (NE) and evaluates the depicted results. Finally, in section 6, the conclusions and the future work are presented.

## II LITERATURE REVIEW AND RELATED SYSTEMS

### Literature Review

Nowadays, networks are very complex systems comprising routers, switches, hubs, and servers that connect numerous devices to crucial applications and through the internet. This complexity grows day by day (Hein, 2019). This means that potential problems must be continuously and effectively monitored before they cause production issues (Guru99, 2019; Xie, 2020). NM is the crucial continuous process of collecting information about network traffic and the state of the various devices (Mohammed, 2013; Parandehgheibi, 2019). According to the OSI model, smart network devices provide an analysis of the network traffic at a first level. At this level, the analysis is limited to physical network problems such as the link status, CRC errors, bipolar violations, and framing errors (Engineering Institute of Technology, 2019).

In the data link, network and transport layers, special monitoring systems are often used to analyse the traffic properties (Marques et al., 2019). Such systems are often referred as “protocol analysers” because special protocols are utilized to check the status of the data transmissions. The most widely-used protocol is SNMP (Simple Network Management Protocol), due to its compatibility with a variety of different products and due to the fact that it is free (SolarWinds Worldwide, 2019). It should be underlined that the latest generation of NM products has been designed to support specific management applications. For example, some monitoring products, like the exinda SD-WAN, have been designed to provide the network monitoring personnel with real-time management information. In addition, other products have been designed to analyse the performance of specific applications and/or to collect data for further analysis such as the exinda Network Orchestrator (GFI Software, 2020). In any case, the companies that operate networks must have installed appropriate and effective NM tools along with the necessary security ones (Aaron, 2015).

### Related Systems

There exist many popular SNMP monitoring tools, such as the following: SolarWinds Network Performance Monitor, Paessler PRTG Network Monitor, Kaseya Network Monitor, SysAid Monitoring, Pulseway IT Management Software, Atera, Spiceworks Network Monitor, and Ipswitch WhatsUp Gold (Guru99, 2019; SolarWinds Worldwide, 2019). These monitoring tools provide device depiction, network performance, security analysis and topology mapping (Hein, 2019). Since visualization is the key to effective NM, these tools should be able to at least visualize the network end-to-end revealing the origin, the requests, the destination, and the route of data (Hein, 2019). It is obvious that such an information monitoring extends in many layers and covers speed, data packets losses, latency, and throughput (Hein, 2019). In order to properly evaluate such systems, the network administrators should have access to a free fully-functional trial version of the tool to be purchased (SolarWinds Worldwide, 2019).

The rapid advancement of technology gave us information in an instance. Network connection is vital in personal usage as with this connection we may gain an extra edge in knowledge information. With this advancement come a few problems such as spam, virus and etc. Therefore, a solution is needed to prevent those attacks before it happens.

Intrusion can occur internally or externally. An internal intrusion is an intrusion from within own networking system. They have an access to the networking system. It may be a friend, partner, employee, or even disgruntled client. External intrusion as it sounds is an intrusion from outside of the network system. Also known as attack from the internet.

Network-based intrusion detection places sensors inside a private network, between routers or a switch. This breaks up a network into multiple smaller networks. The sensors test programs at the network level, and the sensors recognize the activity of the program as normal or abnormal, based on existing comparison parameters. The sensor determines if the program is from outside the network, and how to treat it if it is. Educating household internet user on the benefit of having an intrusion detection system on top of firewall and antivirus.

### III THE PROBLEM SPACE

Information about the current network devices is important. The network administrator must be fully aware of the services offered in his administration domain, and their status. Alerts should be produced via audio signals, monitors or automatic e-mail responses when a problem arises, or a new application/equipment becomes available online. An alert must include information about the device, the problem, and its cause. It is also essential that the minimum possible number of meaningful alerts be generated for each problem. If access to multiple devices is limited due to a problem, the appropriate alert signals help the administrator diagnose the problem quickly, while the generated process information should be kept to a minimum. The ten most important reasons for using NM tools (Ip switch, 2010; Help Systems, 2017) includes, (a) knowing what is happening in the network; (b) planning for upgrades or changes; (c) quick problem diagnosis; (d) demonstration of what happens;

(e) knowing when to use the backup system; (f) confirmation that the safety

devices work properly; (g) monitoring devices used by the company customers; (h) receiving updates on the network status from anywhere; (i) maintaining the uptime of the client and, (k) saving money.

An NM system must at least include network maps, data reports, alerts, historical information, 24-hours a day and 7-days a week operation, remote access, role-based access control and multi-methods (Tittel, 2017).

It is well-understood that one important factor in the competition among network provider companies is how the NM efficiency can be considerably improved as much. For instance, a company A which has medium qualified staff and monitors its network effectively can produce better products/services more quickly than another company B with a highly qualified staff but with less effective network monitoring. This happens because ineffective network monitoring may not detect the upcoming problems in time. This means that these problems may become more severe and the partially and not accurately informed administrators have to search and interfere in multiple network areas in order to restore the network and/or restore the network-based company's functions. Apparently, this time delay may make the difference in the general company's performance.

#### IV REQUIREMENTS OF SYSTEM FUNCTIONALITY

An NM system should support the network administrator's needs while it aims for optimal performance. Since in a modern network environment, the number of managed telecommunications objects is significantly high, monitoring each one of these objects is virtually impossible. Thus, there is a need for a system that manages all these objects and filters the information that ultimately reaches the administrator, preventing an information "bombardment". Only the highly-deemed, based on their significance, alerts should reach the administrator.

The system support of the graphical simulation of its objects on a two-dimensional map can provide a variety of benefits. The objects should be moveable and easily placed in any position, where they can be saved. The objects can be connected with links and the presentation of their telecommunication contents (cards, ports) has to be supported, too. Each item should contain real-time information about the number of active alerts and their importance. All this information should be available on the user screen. Moreover, the system should support and provide statistics, such as how many times a link or a port is out of order, for the managed telecommunication objects.

It is vital that implementation be done using the latest available technology. In the proposed prototype system, the client-server architecture combined with a relational database is used. The supported telecommunication objects and their contents will be stored in the database and will be accessible to the clients. Java Enterprise Edition (JEE) is used. For the implementation of the SNMP mechanism (Network Instruments, 2005), the SNMP4J library is used (SNMP4J, 2003). The object representation will be developed from the scratch using a Java graphics device (Abstract Window Toolkit and Swing). In Table 1, the various system functionality requirements are presented.

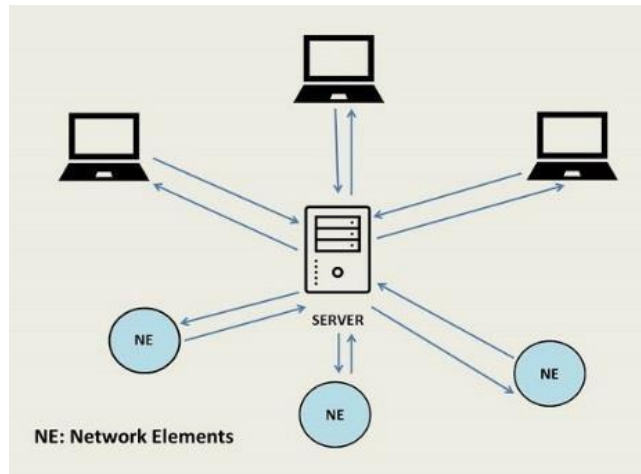
**Table 1 Requirements of System Functionality**

ID	Requirements of System Functionality
R1.	A schematic representation of already joined telecommunication objects/devices on a surface is provided
R2.	Any chosen icon can represent every object type through proper programming
R3.	Each object name/description must be provided
R4.	The background surface map must be given
R5.	The objects can be moved over the surface and their desired position can be saved
R6.	The objects can be joined with a link
R7.	The objects accept SNMP traps (alerts)
R8.	The object alerts use different severity colours, in descending order these colours are: Red (Critical), Orange (Major), Yellow (Minor), Green (Warning)
R9.	The objects have information regarding the status of the active alerts. The number of alerts with the highest severity is displayed. If there are active alerts of lower severity, they are marked with a "+"
R10	In case of a new alert, a "balloon" is displayed over the object which generated it
R11	When the user clicks on the object that contains a "balloon", the alert is acknowledged, and the "balloon" disappears.



## System Architecture

**System Topology and Architecture:** The system aims to provide the user with monitoring services for a managed object/device group. The system topology is depicted in Figure 1 (Network Traffic Monitoring, 2014).

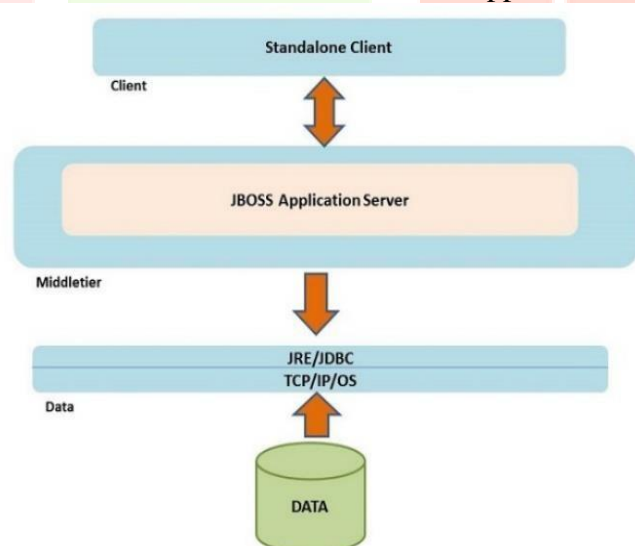


**Figure 1** The system topology

Figure 1 shows that all the network elements are integrated into one server which can perform SNMP operations on the elements. The SNMP operations which can be performed are: SET, GET and TRAPS.

The server can detect TRAPS for events coming from the network elements at a specific port. The network elements should be set on the server that will send the TRAPS, supplying the port and the IP Address of the server. The communication among NEs, server and clients, is interactive.

Here, it should be underlined that the system uses a 3-tier architecture, i.e. clients, application server, and database. JBoss is used as the application server (JBoss Community, 2011) as it can be set up on any machine which runs Java (Java API, 1993). This server can handle several NEs. Finally, several clients, which run on any Java machine, can be connected to the application server (Figure 2).



**Figure 2** The system architecture

The JBoss Enterprise Application Platform (version 4.2.3.GA) is used as a middleware application server, since the JBoss enterprise middleware is widely recognized as one of the most dominant middleware platforms in the market (RedHat, 2019) and it is cost-effective (RedHat, 2019). The examined system uses the PostgreSQL 8.3 database (PostgreSQL, 2009), an advanced, freely available, open source relational (RDBMS) which features foreign keys, sequences, schemas, transactions, triggers, procedure languages, and the BSD – style license.

### Simple Network Management Protocol (SNMP) and Management Information Base (MIB)

SNMP, which is based on UDP (SNMP, 2010), is used primarily in network management systems for monitoring network devices and their reliable operation. SNMP, which is a part of the Internet Protocol (IP) suite, consists of a set of standards for network management, such as an application level protocol, a database scheme, and a set of data objects. SNMP provides data concerning network configuration variables, to the management systems. The values of these variables can be requested or set using the appropriate management applications. Each managed system/element, which is called a “slave”, executes a software component, called an “agent”, that sends information through SNMP to management systems, called “masters”. The SNMP agents send information to the managed systems (slaves) in a variable form. The protocol supports sets, such as change and implementation of a new configuration. The master system can obtain information through the protocol operations GET, GETNEXT, GETBULK or else, the agent, which is placed at the slave, will send data without being requested, using the TRAP or INFORM protocol operations. The INFORM messages of the SNMPv3 are useful because they provide a reliable way of being acknowledged by the management system, a necessity since SNMP is based on UDP. Management systems can send configuration updates through the protocol operation SET to manage a system actively. Configuration update functions are used only when network changes are needed, while monitoring functions are performed daily.

The variables, which are accessible via SNMP, are organized into hierarchies. These hierarchies and other metadata, such as the variable type/description, are described by Management Information Bases (MIBs). SNMP operates at the application layer of the Internet Protocol suite and uses the UDP ports 161 for the agent and, 162 for the management system. The management system can send requests from any available source port to the destination port 161 for the agent. The agent's response will be sent back to the source port. The management system receives notifications (TRAPs and INFORMs) in port 162. The agent can generate notifications at any available port.

SNMP, on its own, does not define which variables a managed system has to offer, because SNMP uses a design which specifies the information within the MIBs. Using ASN1 semantics, the MIBs describe the data structure of a managed system. They also use hierarchical classification and contain Object Identifiers (OID), where each OID defines a variable that can be read and then set via SNMP. The MIB hierarchy is represented as a tree, whose levels are defined by different organizations (domains).

#### Value Objects and Data Access Objects

When they use value objects, the managed systems remain only temporarily in the memory (Figure 3). The business logic is implemented with business components whose life cycle and format are independent from every instance of the telecommunication elements. This approach is consistent with the solution of TMF814, according to which management systems execute business functions and use CORBA structures to represent instances of telecommunication elements.

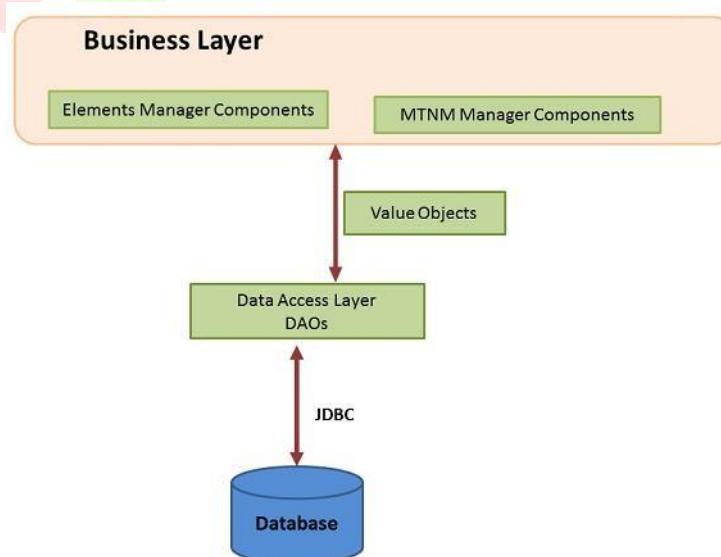
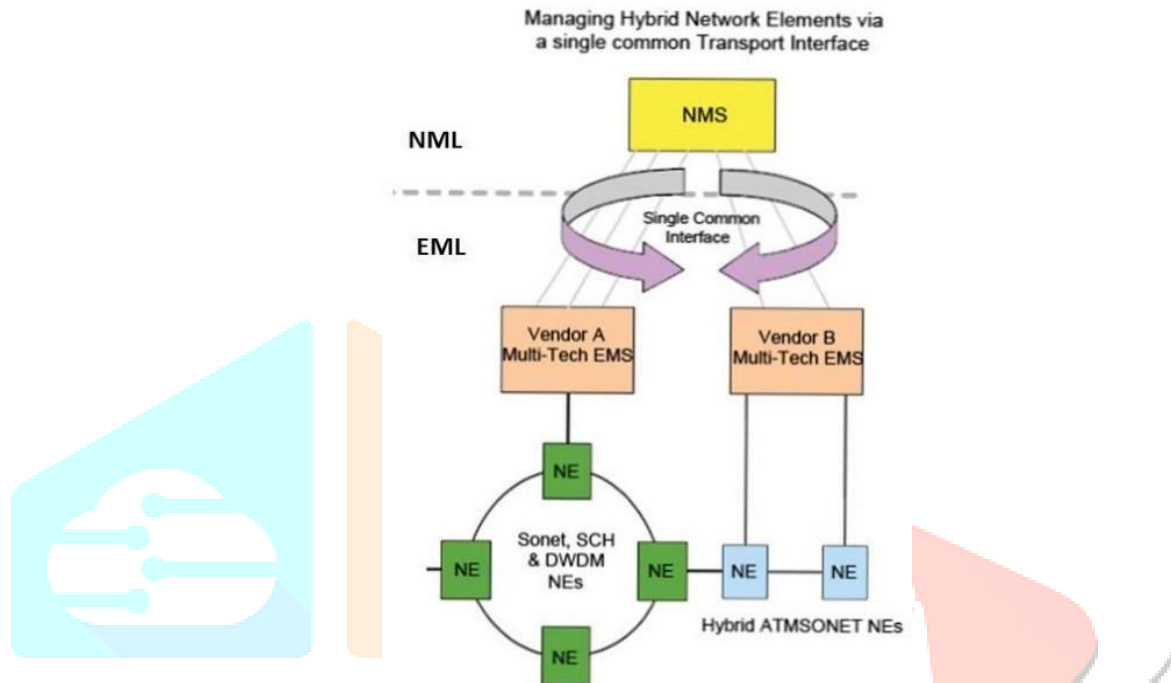


Figure 3 DAOs, Value Objects and Business Layer

The Data Access Objects (DAOs) are classes that contain all the functionality for the execution of SQL statements, and for the creation, the deletion and the modification of rows and tables in the database. The DAOs fill value objects and return them, using JDBC for the direct access to the RDBMS.

### The Multi-Technology Network Management

For the application implementation, the TMF 608 Multi-Technology Network Management (MTNM) NML-EML Interface Information Agreement, Version 3.0 is used. TMF 608 defines the interface between the EML and the NML layers, so that the latter provides Network Management services to network elements which support multiple technologies (Lewis, 1999).



**Figure 4 TMF 608 Management Approach (LIGHTWAVE, 2000)**

As shown in Figure 4, *TMF608* is part of the MTNM Solution Suite 3.0 (TM Forum, 1988) and consists of a set of documents that stipulates the information exchange or the interface between the Network Management Systems (NMS) and the Element Management Systems (EMS), providing the management of SONET/SDH, DWDM and ATM transmission networks (AdventNet TMF EMS, 2013).

The “business agreement” document of TMF608 describes the problem and the solution requirements. The “information agreement” document of TMF608 uses a neutral approach and provides interfaces which satisfy the requirements of the “business agreement”. This document suggests that the problem be solved with a specific technology, namely the Corba IDL. In addition, the implementation model with its instructions provides a mechanism for the compliance of the implemented solution. In brief, the proposed solution of TMF MTNM version 3.0 contains the documents listed in Table 2:

**Table 2 MTNM version 3.0 Specification Documents**

Document	Document Number
Business Agreement	TMF513
Information Agreement (Rational Rose Version)	TMF608
Information Agreement (HTML Version)	TMF608
Solution Set	TMF814
Implementation Statement	TMF814A

As shown, the *TMF MTNM* project (version 3.0) has produced a solution based on CORBA, *TMF814* and *TMF814a*, whereby the IDL interfaces and structures have been defined. A solution based on Java for the MTNM version 3.5 is expected. This will be based on XML and JMS. When such a solution is used, the NML and EML levels communicate with vendors neutrally. This is important for any EMS or NMS which is compatible with MTNM.

### TMF MTNM Information Model Objects

*TMF 608* defines the interfaces that an EMS/NMS system is required to have to meet the MTNM business agreement. Table 3 provides a brief hierarchical description of the major applied model entities and some key variables. It should be noted that these entities are tables in the application database and the variables are columns in these tables. These variables specifically identify each entity using a unique identifier (name = unique id in DB), related names (EMS) and user-friendly representations (nativeEMSname, userLabel).

**Table 3 Model Entities**

Number	Model Entities
1.	EMS
2.	Subnetwork
3.	ME
4.	Equipment Holder
5.	Equipment Holder Slot
6.	Equipment
7.	PTP
8.	TL

The subnetworks do not represent a natural resource, but rather a type of network topology. A subnetwork is contained in an EMS and includes TLs (topological links), MEs (managed elements), TPs (termination points) as well as SNCs (subnetwork connections). The purpose of subnetworks in MTNM is the creation and deletion of SNCs. An SNC ends at CTPs (Connection Termination Points), which are contained in the MEs of the object "subnetwork".

The subnetworks have a "layer rates" variable which represents the layer rates supporting SNCs. An ME is not allowed to be a member of two subnetworks within the same layer rate.

The Element Representation Library (ERL) is a set of classes which is compatible with MTNM version 3.0 and helps an external programmer to simulate objects on a 2D flat surface. Even though ERL was created for telecommunications, it was designed in such a way that the modelling of any included graphical representation is facilitated. This means that the programmer does not need to provide some type of extra library to run the ERL but only a code, which is able to initialize and "bind" the function of each class with the rest of the system. The following sub-sections describe the capabilities of this library, its main components, and its applications.



## System Design and Development

The motivation for the development of ERL was the fact that all the other implementations were either tied to a specific platform or they were so complicated that only 10% of their classes are used.

A considerable effort has been put to maximize the range of the ERL applications. ERL can create a model for a variety of applications, from railway and airport networks to educational programs for children. However, it has been mostly developed for modelling telecommunication networks.

Even though the entire range of applications may seem diverse, all the applications include the major elements of ERL, like surface, background, and represented items. ERL can also be used both as a stand-alone application and as a sub-module within large applications that require a graphical representation.

When it is used in a network (rail, roads, telecommunications, satellite, or air), the main reason for creating such a representation is to give to the user an overall view of an object set interaction. This helps considerably in remote management and control. For example, in a chess game, such a representation would provide the main interface where the user would interact and play. The idea is since people understand graphics more easily and faster than text. ERL provides the means which help us display very complex information. Moreover, it is very accurate because it uses double precision numbers to position an object. Furthermore, the use of ERL is easier than any other interface and requires minimal training. The following

sub-sections describe the ERL structure. The main items, which are called elements, are hereafter presented briefly as to what type of objects they are.

### Graphical Objects

The graphical objects, as described in Table 4, are virtual elements that are visible to the user.

**Table 4 Graphic Objects**

Object s	Signature	Description
GObject → GObject	-	Virtual entity that extends to GObject
GObject	GObject (String id, String name, Boolean movable, Boolean selectable)	Contains two instances of the class ERL Alarm Counter(alarm Counter, new Fault State Values Counter)
GBO_Netwo rk Element	GBO_Network Element (String id, String name, Boolean movable, Boolean selectable, Point2D DoublereferencePoint, ImageIcon icon)	The most important depicted object in the ERL. It looks like an icon on the PC desktop. It takes a complicated form with added states. It inherits from GObject.
GBO_Link	-	It is a straight line from an object centre to another object centre. It demonstrates a direct relationship between them. It can be internal or external according to the programmer. It is not a moving object. It just follows the moving objects. In Java, it is recalculated and redisplayed fast and precisely using Bresenham's Algorithm (Buss S. R., 2019). Its default density is 5 pixels. When selecting a link, a white line that surrounds it can be seen. This line is depicted differently depending on the link state. In some cases, it will appear as a 3D picture and in others as a shaded red line. A triangle in the centre represents old (coloured area) and new fault states (coloured interior triangle) until acknowledged by the user. If there is no fault state,

then there is notriangle.

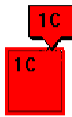
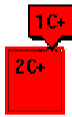
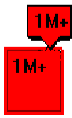
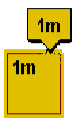


A GObject (Graphical Object) is the basic class of all the objects displayed using ERL. Anything that is displayed must inherit from this object. The ERL library heredity extends from the GObject to the GBOobject (Graphical Business Object). This happens because the most important need in the use of an ERL is to have ready objects that can be used immediately.

As far as the existing alarms are concerned, the provided information contains three characters. The first describes the number of alarms with the most important severity. The second may be one of the characters (C, M, m, W, i), which stand for Critical, Major, minor, Warning, indeterminate. Finally, the third, which may or may not be displayed, is the + character, which indicates that other alarms of lesser severity exist. The way all this information is displayed depends on the GBOobject type and is implemented with the paint() method.

The new alarms are described in the same way as the existing ones. The difference lies in the fact that these alarms arrive on a run time basis and have not been acknowledged yet. They are displayed through a balloon, coloured with the severity colour, containing the above described information, and placed over the GBOobject they relate to.

**Table 5 shows some alarm statuses, while tables 6 and 7 contain all the used methods and all the ERL elements, respectively.**

**Table 5 Some alarm statuses**

Alarm Status	Visual	Comment
New Critical		The resource has one new critical alarm.
Outstanding Critical		The resource has one new critical alarm, plus less severe new alarms, and one acknowledged critical alarm.
New Major		The resource has one new major alarm, plus less severe new alarms.
New Minor		The resource has one new minor alarm.
Acknowledged Minor		The resource has two acknowledged minor alarms, plus acknowledged less severe alarms.
Outstanding Warning		The resource has three warning alarms, plus two acknowledged warning alarms.

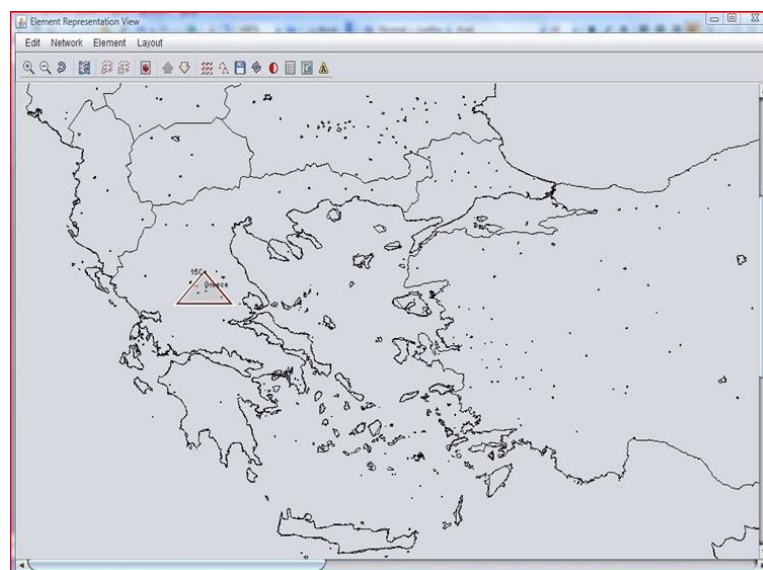
**Table 6 Methods and their description**

Methods	Description
paintAlarmBallon()	It represents the balloon over the GBOject when new alarms occur.
addChild()	It fills the vector children of the GBOject class through a GBO_Link.
times_linked_together()	It returns the number of links connecting two GBOjects and is used by the GBO_Link class.
paint()	It is implemented by the GBOjects. It runs through the objects in the vector level objects which are displayed in an ascending vector order (0, 1, 2, etc.). It is overridden by ERL View for displaying custom graphics on the canvas.
contains()	It is implemented by the GBOjects. It finds out if any object contains the clicked point. If not, it returns null as the GObject.
setPrimaryState()	It sets the primary object status and changes the current value of ERL Primary State Carrier together with methods that return and reflect the primary condition.
getReferencePoint()	It gets the reference point of the movable objects.
move()	It changes the reference point of the movable objects through the user's drag and drop functionality.
combineVerticalDimensions()	It changes the length of the glyph icons vertically (north and south orientations).
combineHorizontalDimensions()	It changes the length of the glyph icons horizontally (east and west orientations).

## V OPERATIONS AND EVALUATION

### General Operations

This section provides a demonstration of how the prototype application works. A proper installation is very important. In this system demonstration, three NEs have already been added. The elements are connected by links and have a set of alarms. For causing SNMP Traps, several shareware tools, such as I-Reasoning and MG-Soft can be used. Here, I-Reasoning is used. This tool facilitated the sending of Traps: linkUp, linkDown. The port to which Traps will be sent is 8088. The community has the form sa.public.<slot\_number>. The variable bindings should have ifIndex 6-77 (corresponding to 6 at the door 1, 77 at the door 72). Figure 7 shows the region which the system monitors.



**Figure 7 Home screen – region**

A map of the supervised region and an object are displayed. On the object, the name, and the number of all the alarms for its subobjects are shown. In this case, the object has 15 critical alarms and the symbol “+” informs us that there are many other alarms with lower severity. Its toolbar contains several function buttons (Table 10).

**Table 10 Function Buttons**

Number	Buttons
1.	Zoom (Zoom In)
2.	Decrease (Zoom Out)
3.	Restore the objects in their original dimensions
4.	Show/Hide map levels
5.	Select all objects
6.	Deselect all items
7.	Move levels (Up/Down)
8.	Order of objects
9.	Automatic layout of objects
10.	Save the positions of objects
11.	Refresh
12.	Enable/Disable blinking
13.	Port Statistics
14.	Element Statistics
15.	Active alarms

## VI CONCLUSION

In this paper, we studied network monitoring systems and highlighted their importance through a detailed analysis, design, and description of a prototype network monitoring system. The knowledge of what is happening in a network every moment with alarms, performance, and status, is vital with respect to efficient network administration.

The proposed system provides a graphical representation of the telecommunication elements/objects and their content on a surface. These telecommunication objects accept SNMP traps, the server “listens” to these traps and the client simulates them in an alarm form at the corresponding object each time. The user acknowledges the active alarms. The alarms are stored in a database and are available to the client through a corresponding screen. In addition, each time a network link is down, the corresponding counter comes up in the database. These statistics and other information can be available to the user through the appropriate administrator interface.

As network monitoring tools are continually developed and improved to satisfy the growing needs of the telecommunications operations sector, there are still some issues, which are open to be investigated in future. These include, among others, the tree representation of the inserted objects, a graphical display of the collected statistics and an automated way to import objects into the system. Moreover, data analytic techniques and predictive algorithms based on machine learning techniques may greatly enhance the system performance and the provided services (Ayoubi et al, 2018).



**VII REFERENCES**

- 1) Aaron K. (2015) "United Airlines Needed Simple Network Monitoring", Ipswitch Blog, [online], <https://bit.ly/3lfuSdQ> (Accessed on 21st April 2017)
- 2) AdventNet TMF EMS (2013), Manage Engine, [online], <https://bit.ly/3ntvsGG> (Accessed on 8th May 2017)
- 3) Ayoubi S. et al (2018) "Machine Learning for Cognitive Network Management", IEEE Communications Magazine, 56(1), January 2018.
- 4) Buss S. R. (2019) "3D Computer Graphics "A Mathematical Introduction with OpenGL", [online], <https://bit.ly/30GdPd8> (Accessed on 14th December 2019)
- 5) Ciuffoletti A. and Polychronakis M. (2012) "Architecture of a Network Monitoring Element", [online], <https://bit.ly/2GqwxYJ> (Accessed on 21st April 2017)
- 6) Engineering Institute of Technology (2019) "Practical TCP/IP and Ethernet Networking for Industry", [online], <https://bit.ly/3d3W7r7c> (Accessed on 13th December 2019)
- 7) GFI Software (2020) "Network Monitoring Tools", GFI, [online], <https://bit.ly/3iDSWWb>, [online], (Accessed on 21st April 2017)
- 8) Guru99 (2019) "43 BEST Network Monitoring Tools [Dec 2019 Update]", <https://bit.ly/30HbSNu> (Accessed on 2nd December 2019)
- 9) Hein D. (2019) "Why Is Visibility Important for Network Monitoring?", Network Monitoring Solutions Review, <https://bit.ly/2Sw8uAC> (Accessed on 2nd December 2019)
- 10) HelpSystems (2017) "Top 7 Benefits of Network Monitoring Stay ahead of outages, fix issues faster, and get immediate ROP", <https://bit.ly/3ddpsgH> (Accessed on 30th November 2019)
- 11) Ipswitch (2010) "The Value of Network Monitoring", Techdata, [online], <https://bit.ly/2Gqx6sl> (Accessed on 21st April 2017)
- 12) Java API (1993) Oracle, [online], <https://bit.ly/2Gyi2J1> (Accessed on 3rd April 2017)
- 13) JBoss Community (2011) "Community driven open source middleware", JBoss Developer [online], <http://www.jboss.org/> (Accessed on 3rd April 2017)
- 14) Lewis D. (1999) "A Development framework for open service management systems", [online], <https://bit.ly/2I5wkRT> (Accessed on 8th May 2017)
- 15) Lightwave (2000) "Major vendor cooperation yields network-management interoperability", [online], <https://bit.ly/2SA1Bkg> (Accessed on 8th December 2019)
- 16) Marques J. et al. (2019) "An optimization-based approach for efficient network monitoring using in-band network telemetry", Journal of Internet Serv. Appl., 10, 12 (2019).
- 17) Mohammed Sb A. (2013) "Network Traffic Analysis: A Case Study of ABU Network", [online], <https://bit.ly/33Dkgja> (Accessed on 8th December 2019)
- 18) NET-SNMP (1992) [online], <http://www.net-snmp.org/> (Accessed on 21st April 2017)
- 19) Network Instruments (2005) "SNMP Monitoring: One Critical Component to Network Management", Viviani Solutions, [online], <https://bit.ly/2F8XbeJ> (Accessed on 8th May 2017)
- 20) Network Traffic Monitoring, (2014), Topology, [online], <https://bit.ly/3noXxPF> (Accessed on 21st April 2017)
- 21) Parandehgheibi A. et al. (2019) "Augmenting flow data for improved network monitoring and management", United States Patent, January 8, 2019
- 22) PostgreSQL (2009), [online], <http://postgresql.gr/node/3> (Accessed on 8th May 2017)
- 23) RedHat (2019) "What is middleware?", [online], <https://red.ht/3nsUZA7> (Accessed on 13th December 2019)
- 24) SNMP (2010) "SNMP Monitoring and Management", Oracle [online], <https://bit.ly/36AS16y> (Accessed on 21st April 2017)
- 25) SNMP4J (2003) "Free Open Source SNMP API for Java", snmp4j, [online], <http://www.snmp4j.org/> (Accessed on 21st April 2017)

- 26) SolarWinds Worldwide (2019) "12 Best SNMP Monitoring Tools + Complete SNMP Guide", [online], <https://bit.ly/30JoSCI> (Accessed on 13th December 2019)
- 27) Tittel E. (2017) "How to select the best monitoring tool", [online], <https://bit.ly/2SwB6tF> (Accessed on 8th December 2019)
- 28) TM Forum (1988) "TM Forum MTNM Project", TMForum [online], <https://www.tmforum.org/> (Accessed on 21st April 2017)
- 29) Xie K. et al. (2020) "Accurate and Fast Recovery of Network Monitoring Data With GPU-Accelerated Tensor Completion," IEEE/ACM Transactions on Networking, vol. 28, no. 4, pp. 1601-1614, August 2020

