Volume 5, No. 3, March 2014
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

CONCEPTUAL ONTOLOGICAL MODEL FOR PRIVATE ENTERPRISE MIB UPDATE

Vinod Kumar Shukla*[1], Dr. D.B. Ojha*[2]

*[1]Research Scholar, Mewar University, Chittorgarh, Rajasthan, India
vinod_shukla@ymail.com
*[2]Professor, Mewar University, Chittorgarh, Rajasthan, India
[2] *ojhdb@yahoo.co.in*

Abstract: Internet growth has become the challenge for the network management. Our paper is divided into two sections. Section –I is introduction. Section – II is based on the concepts of SNMP, MIB, OID and Ontology. In this section there is discussion about our proposed method in which, we have taken a model of manager and agent communication. For efficient network management there is a need for continuous update of network devices with the latest statistics, on which they operate. In network, the devices which work on the concept of SNMP use the MIB values, which are updated by exchanging OID values to update network themselves. If at any point of time, there is a need to add new OID in MIB tree by the private vendor then this could be done with the help of Ontology. We have proposed the concept of Onto-Agent, which will work on all the managed devices along with the manager. On manual update of new OID in manager, onto-agent will also read this value and generate the ontology based on the new updated OID in MIB tree and will communicate this to all the connected managed devices, on the receiving end onto-agent will receive this and convert this ontology into updatable OID in MIB. From this process all the managed devices will update this new OID into their MIB.

Keywords: SNMP, MIB, OID, ASN, Ontology, OWL-DL

INTRODUCTION

With the immense growth of Internet users, Network management has become an issue and a challenge to implement. As network is growing the infrastructure to support in terms of hardware and software is also increasing day by day. To maintain, update and manage this big network is a challenging task for the network management companies. Due to this, the need for efficient management of network resources has emerged as an alarming issue. When there is a need to update or maintain the network objects we need some mechanism which is faster and efficient for the similar/dissimilar network.

The network and network device which works on the concept of MIB, they keep on exchanging the values of MIB's OID for the current status. In our paper, we have proposed a conceptual ontological system for private MIB updation, which help to update each device in the network. In this proposed system, to update all the devices in network in efficient and fast manner, ontology will use the MIB's OID values.

SNMP

SNMP works on the concept of request/response. All the network management systems send out a request and the managed devices return a response. This is implemented using one of four operations: Get, GetNext, Set, and Trap. SNMP messages consist of a header and a PDU (protocol data units). The headers consist of the SNMP version number and the community name. The community name is used as a form of security in SNMP. [1]

When an SNMP device sends a trap or other message, it identifies each data object in the message with a number string called an object identifier, or OID.

Up to now, more than 100 Request for Comments (RFC) documents, related to MIBs, have been proposed [2] and at least 15 RFC documents, with the "standard" or "proposed standard" status, are related to host resources.

MIB

A MIB (Management Information Base) is a text file which has been written using the ASN.1 (Abstract Syntax Notation) format. This text file is human readable but is special in that it can be compiled by a computer program called a MIB compiler, and then will result in creation of objects called OIDS (Object Identifiers), that can be understood by a network management station using the SNMP (Simple Network Management Protocol) method of communication. Why is this important? SNMP MIBs are crucial in order to manage your network and understand the underlying objects which are being retrieved from SNMP Agents. [3]

The MIB is an ASCII text file that describes SNMP network elements as a list of data objects. The most widespread of these, MIB modules, are used mainly for monitoring the state. If a device at any time fails to respond to a query about its state, then no information about its current state or behavior can be decided.

Manufacturers can also provide custom MIB files for their hardware. Some of the features which are not available in the MIB can be provided by the vendor. The custom MIB files are loaded in to NMS to tell the software which SNMP objects can be queried, and how to display the results.

MIB files themselves are difficult to read, they are only meant to be imported, or "compiled" by a Management Station. [4]

Private vendor must have the complete list of manageable object as a design what they want to include in the developing MIB. After developing MIB register the private MIB branch by obtaining a MIB registration point from the IANA (Internet Assigned Numbers Authority). After registering this private MIB branch, arrange each manageable object in a tree-like structure, and assign an OID to each variable, creating sub trees for each manageable object group. [5]

OID

OIDs or Object Identifiers uniquely identifies managed objects in a MIB hierarchy. This can be depicted as a tree, the levels of which are assigned by different organizations. Top level MIB object IDs (OIDs) belong to different standard organizations. [1]

While each OID is unique, the first several pieces of each OID are almost always the same. These upper location levels are defined by a series of standard reference within the MIB. These series are called RFCs, or Requests for Comments. The RFCs that define SNMP OIDs are part of a larger group of RFC documents that define the Internet as a whole. Individual vendors create their own MIBs that only include the OIDs associated specifically with their device. [6]

OIDs are very structured, and follow a hierarchical tree pattern. The top level after the root is ISO, and has the number "1". The next level, ORG, has the number "3", since it is the 3rd object under ISO, and so on. OIDs are always written in a numerical form, like the top three object levels are written as 1.3.1.

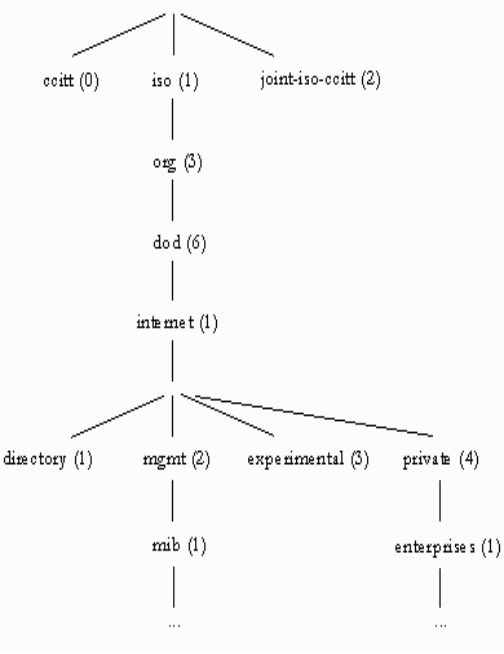

Figure 1: Hierarchy of MIB Tree

A SNMP OID functions as an address that identifies the location of a specific element within the entire SNMP network. The translation of OIDs allows the SNMP manager to determine values for these objects. The MIB assigns readable labels to each OID, which allows the manager to interpret and assemble SNMP messages. Without the OID, the message cannot be translated into a form that is readable to humans.

When the SNMP manager requests the value of any object, it assembles a message with the OID, which is sent to the MIB for decoding. If the OID is listed within the MIB at that particular management station, a message is sent back to the manager including the value requested for that particular OID.[6]

OID are the unique values, given to objects into MIB tree. All child nodes are given unique integer values within that new sub-tree. Children can be parents of further child sub-tree (i.e. they have subordinates) where the numbering scheme is recursively applied. Children of the same parent cannot have the same integer value. [7]

ASN.1

ASN.1 is a formal notation used for describing data transmitted by telecommunications protocols, regardless of language implementation and physical representation of these data, whatever the application, whether complex or very simple. [8]

The MIB is written in ASN.1 notation. (The initials stand for Abstract Syntax Notation 1.) ASN.1 is a standard notation maintained by the ISO (International Organization for Standardization) and used in everything from the World Wide Web to aviation control systems. ASN.1 is human readable and specifically designed for communication between computer systems. Abstract Syntax Notation One (ASN.1) [9] is a framework for representing tree structured data.

As well as the definitions of management information contained in the MIB, the SMI standard defines the rules used to define and identify these variables. This restricts the type of variables allowed in the MIB, specifies the rules for naming these variables and creates rules for defining the variable types.

The SMI standard specifies that all MIB variables must be defined using ISO's Abstract Syntax Notation 1 [ASN.1] which is a formal language allowing a hum an readable form as well as a compact encoded representation which can be used in communication protocols and prevents any ambiguity in the form or content of any variable. [10]

ONTOLOGY

An ontology is a representation or model of knowledge, a "formal, explicit specification of a shared conceptualization" according to (Gruber, 1993), and this means that however 'shared' it may be it is still extremely subjective, representing the time, place and cultural environment in which it is created.

Ontology refers to the interpretation of a group of ideas within a specific domain that defines the interrelationship

between those ideas. Ontology can be used to study the existence of entities within a specific domain and sometimes can be used to identify the domain itself. [11]

The advantage of ontology is that it represents real world information in a manner that is machine process able. The reason ontologies are becoming popular is largely due to what they promise: a shared and common understanding of a domain that can be communicated between people and application systems. [12]

OWL AND OWL-DL LANGUAGE

The OWL Web Ontology Language is an international standard for encoding and exchanging ontologies and is designed to support the Semantic Web. OWL is an ontology language for the Web. It became a World Wide Web Consortium (W3C) Recommendation in February 2004. As such, it was designed to be compatible with the eXtensible Markup Language (XML) as well as other W3C standards. [13]

OWL-DL is grounded on Description Logics, and focuses on common formal semantics and inference decidability. Description logics offer additional ontology constructs (such as conjunction, disjunction, and negation) besides class and relation.

The strong Set Theory background makes Description Logics suitable for capturing knowledge about a domain in which instances can be grouped into classes and relationships among classes are binary. OWL-DL uses all OWL ontology constructs with some restrictions.

PROPOSED SYSTEM

In this paper we have proposed a concept of MIB auto update on various connected managed objects with the help of ontology. As we know that all the managed objects need to implement the concept of MIB in order to achieve proper coordination with all the other connected managed objects and as well as with proper communication with SNMP manager.

Communication between SNMP manger and all other managed objects is done by exchanging the values of MIB, with the help of SNMP trap messages. A SNMP trap message contains the OID values. OID uniquely identifies managed objects in a MIB hierarchy.

In MIB hierarchy, vendors define private branches including managed objects for their own products. If at any point of time any update needs to be done by a private vendor into its own MIB in all managed devices, then it can be done by centrally updating at Manager Level. One separate application onto-agent is working in all the managed objected including manager.

Manager onto-agent will understand this new MIB update and will generate the ontology for the new update only and communicate to all connected managed object. Onto-agent will understand the MIB notation (OID values) and convert this into XML based ontology.

All other managed object's onto-agent will receive this ontology and translate into OID which will be updated into their respective MIB.

Onto-Agent is a proposed concept, in this paper, which will work on all the managed objects along with the manger. This will be responsible for the understanding ontology sent by all the other managed devices as well as manager. And this will also help to generate the ontology from the MIB OID values.

Figure 2: Implementation steps for proposed system. Resours shown in figure are different managed objects.

As per the proposed Ontological IDS system on [14], let's take an example that one more property need to be added and we need to modify the existing MIB of IDS and at the same time it need to be updated in central class of IDS and subsequently this need to be updated in HIDS and NIDS.

An ***Intrusion detection system (IDS)*** inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. [15]

Host-based IDS - A *Host-Based IDS* monitors the activity on individual systems with a view to identifying unauthorized or suspicious activity taking place on the operating system.

Network-based IDS - A *Network-Based IDS* is solely concerned with the activity taking place on a network (or more specifically, the segment of a network on which it is operating). [16]

Supposed IDS MIB is owned by some private owner and its MIB's OID is 1.3.6.1.4.1.XXXX.1. And it has already three property defined, Traffic collector, Analysis engine, Reporting method. (We have taken an example of XXXX as registered MIB for IDS from private vendor.)

Respective OID's managed by private vendor are 1.3.6.1.4.1.XXXX.1.1 (for Traffic Collector), 1.3.6.1.4.1.XXXX.1.2(for Analysis Engine), 1.3.6.1.4.1.XXXX.1.3(for Reporting Methods) , and let's take the example that one hierarchy (a new OID in MIB of IDS after registration) need to be added is PACKET STATUS in node of XXXX, which has the attribute like, PacketIndex, PacketMidifcationStatus, PacketCPUUses and PacketSize defined below. (In this example XXXX is new OID assigned to IDS MIB).

According to above stated OID distribution we can summarize the different levels of Packet Status (OID: 1.3.6.1.4.1.XXXX.1.4) into following:

Table 1: Description of New OID in MIB

Leaf OID Index	***Description***
PacketIndex - (1)	Index
PacketModificationStatus (2)[1]	Date
PacketModificationStatus (2)[2]	Time
PacketCPUUses- (3)	CPU utilization
PacketSize - (4)	In Bytes

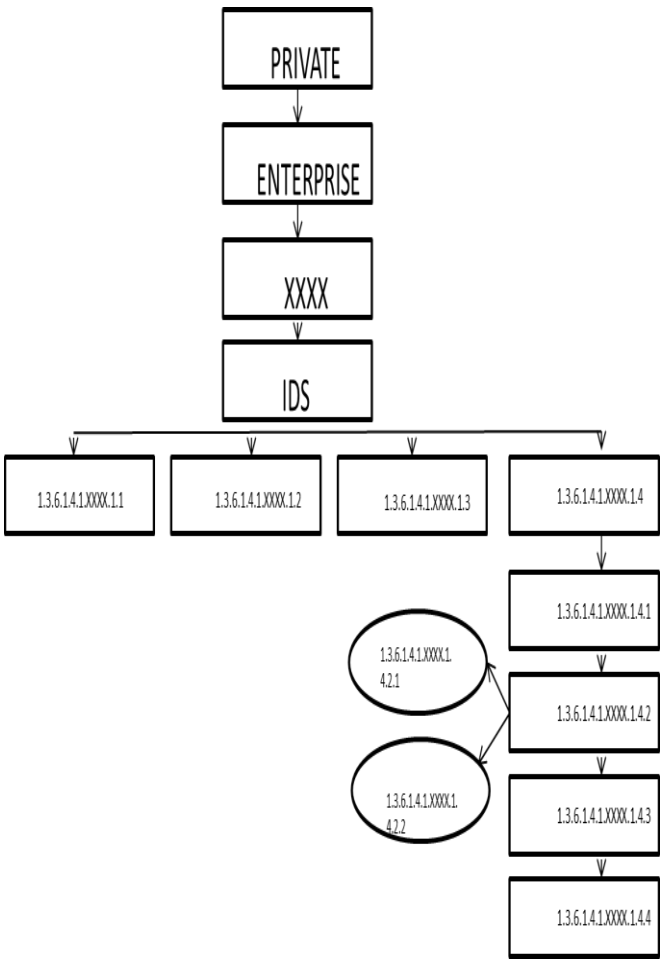

Figure 3: Hierarchical levels of Private MIB (XXXX)

In Figure:3 XXXX is a registered number by private vendor in MIB tree hierarchy. XXXX is owned by private vendor for IDS, which can be further classified in other levels. Here we have taken the example in which XXXX is one of the classification is for IDS and we have assigned different levels according to class defined in our proposed model for IDS. [14]

Suppose one more common component Packet Staus is added into the IDS class and given the OID number XXXX.1.4 which is further broken into other leaf process shown in Figure: 3 and illustrated in Table:2.

Table 2: Description of hirerachy of new added OID in MIB tree

Leaf MIB	***Leaf MIB's OID***
PacketIndex	1.3.6.1.4.1.XXXX.1.4.1
PacketModificationStatus _Date	1.3.6.1.4.1.XXXX.1.4.2.1
PacketModificationStatus _Time	1.3.6.1.4.1.XXXX.1.4.2.2
PacketCPUUses	1.3.6.1.4.1.XXXX.1.4.3
PacketSize	1.3.6.1.4.1.XXXX.1.4.4

After the new OID are assigned, this new MIB need to be updated. New MIB will be updated in Manager manually by vendor. In manager onto-agent will also read these new updated values from MIB database and will generate the new ontology for recent defined OID. For example the Packet Status is defined in OWL-DL language as following.

```
<owl:ObjectProperty rdf:ID=" hasPacketIndex ">
<rdfs:domain rdf:resource="#IDS"/>
<rdfs:range rdf:resource="# PacketIndex "/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID=" hasPacketModificationStatus ">
<rdfs:domain rdf:resource="#IDS"/>
<rdfs:range rdf:resource="# PacketModificationStatus "/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID=" hasPacketCPUUses ">
<rdfs:domain rdf:resource="#IDS"/>
<rdfs:range rdf:resource="# PacketCPUUses "/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID=" hasPacketSize ">
<rdfs:domain rdf:resource="#IDS"/>
<rdfs:range rdf:resource="# PacketSize "/>
</owl:ObjectProperty>
```

This ontology for the new added OID will be sent to all connected NIDS and HIDS, which are also running onto-agent with them. Onto agent will read this new sent ontology and translate this into the OID values with description and will update their existing MIB database. And the entire connected managed object can update the new added MIB.

CONCLUSION AND FUTURE SCOPE

As the network is growing continuously and to achieve automatic network update this proposed concept can speed up the task. Although so many updation needs to be done in the same direction to achieve this. As we also need to design the complete system for the working mechanism of Onto-Agent, that how Onto-Agent will understand and generate

the ontology for each managed devices. This leaves lot of scope for the further discussions.

REFERENCES

[1] ”PAESSLER KNOWLEDGE BASE”, http://www.paessler.com/knowledgebase/en/topic/653-how-do-snmp-mibs-and-oids-work

[2] RFC INDEX, http://www.ietf.org/iesg/1rfc_index.txt

[3] OIDVIEW, http://www.oidview.com/mibs/detail.html

[4] “SNMP Tutorial Part 2: Rounding Out the Basics”, AARON LESKIW , http://www.networkmanagementsoftware.com/snmp-tutorial-part-2-rounding-out-the-basics

[5] “Developing a MIB”, http://msdn.microsoft.com/en-us/library/aa909833.aspx

[6] SNMP OID, http://www.dpstele.com/dpsnews/techinfo/snmp/snmp_oid.php

[7] “OID - Object Identifier”, http://www.ossexperience.com/SNMP-OID.html

[8] “Introduction to ASN.1”, http://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx

[9] ITU-T. X.208: Specification of Abstract Syntax Notation One, 1988. http://www.itu.int/itudoc/itu-t/rec/x/x200-499/x208.html

[10] SANS Institute InfoSec Reading Room, “SNMP and Potential ASN.1 Vulnerabilities”

[11] Computer Ontology, http://www.techopedia.com/definition/591/computer-ontology

[12] Ontologies Advantages, http://mecca.noc.uth.gr/ontologies_advantages.htm

[13] Chapter 2, “AN INTRODUCTION TO THE OWL WEB ONTOLOGY LANGUAGE”, Jeff Heflin.

[14] Vinod Kumar Shukla and D.B Ojha-Ontological IDS Monitoring on Defined Attack. International Journal of Science and Research, 3(3), 2014, Page 665-670.

[15] What is Network Intrusion Detection System, http://www.combofix.org/what-it-is-network-intrusion-detection-system.php

[16] Intrusion Detection System, http://www.techotopia.com/index.php/Intrusion_Detection_Systems

Dr. Deo Brat Ojha, Ph.D from Institute of Technology, Banaras Hindu University, Varanasi (U.P.), INDIA. His research field is Optimization Techniques, Functional Analysis & Cryptography. He is Professor at Mewar University Chittorgarh Rajasthan INDIA. He is the author/co-author of more than 250 publications in International/National journals and conferences.

Mr. Vinod Kumar Shukla received the degree of MCA from U.P. Technical University in 2004, has total experience of nine years in teaching and training. He is currently pursuing PhD from Mewar University, Rajasthan, India in the area of Semantic web and Ontology