An Ontology Approach to Construction of Identification Subsystem for Intrusion Handling Systems in Wireless LANs

S. Salekzamankhani, A. Pakštas and B.Virdee

***Abstract*- Intrusions to WLANs is a fact of everyday life and handling them is becoming more and more challenging. Currently there is no standardized reference model which can be used to compare and evaluate existing or design future Intrusion Handling Systems for WLANs. Hence this paper describes and discusses the construction of Identification Subsystem Modelling Ontology (ISMO) of Intrusion Handling System's reference model. The proposed ontology is based on the concepts of various ontology modelling and simulation tools. Careful attention is given to support two important functions: manage the dependencies between ontologies and at the same time to keep and restore their consistencies if they change in order to accommodate new information, or to adjust the representation of the domain as the world changes.**

***Index Terms*- Intrusion Detection/Prevention Systems, Ontology, Reference Model, Wireless Networks.**

I. Introduction

WLANs are different from the traditional wired LANs in terms of their exposure to potential threats, vulnerability and security techniques. Hence there is an urgent need for an effective Intrusion Handling Systems (IHSs[1]) to reduce/eliminate such threats.

Analysis of the commercial IHSs shows that they all are built as a proprietary systems which are neither taking into consideration existence of other IHSs nor they are trying to find the ways to establish inter-IHS collaboration which may help to achieve better security for the end-users [1,2]. Fig. 1 shows the proposed IHS reference model in [1,2] which consists of following: identification subsystem, response subsystem, inter IHS communication subsystem, management console, source data, inter IHS communication systems, etc. The identification subsystem's structure is the focus of this paper. An ontology approach will be taken to describe its building block and its structure.

In recent years the development of ontologies (explicit formal specification of the terms in the domain and relations among them [3]) has been moving from the realm of the Artificial-Intelligence laboratories to the desktops of domain experts. Many disciplines now develop standardised ontologies which can be used by domain experts to share and annotate information in their fields [4].

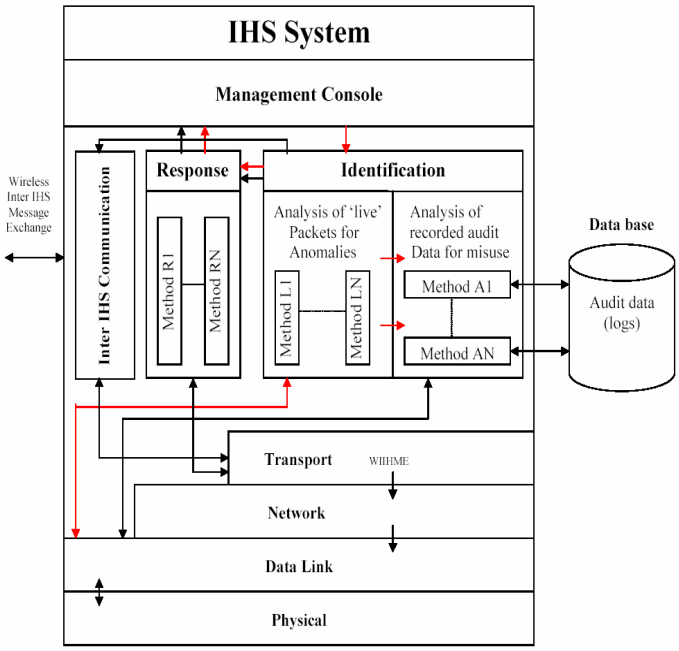


FIG. 1. ARCHITECTURE OF IHS REFERENCE MODEL

Some of the main reasons of ontology development [4] include:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyse domain knowledge

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies [3,5].

Manuscript received February 17, 2010.

S. Salekzamankhani is with Faculty of Computing, London Metropolitan University, London, N7 8DB, UK (Phone: 020 7133 2207, email: s.salek@londonmet.ac.uk).

A. Pakštas is with Faculty of Computing, London Metropolitan University, London, N7 8DB, UK (email: a.pakstas@londonmet.ac.uk)

B. Virdee is also part of Faculty of Computing, London Metropolitan University, London, N7 8DB, UK (email: b.virdee@londonmet.ac.uk)

[1] In this document, a general term "Intrusion Handling System" (IHS) is used as discussed in [1,2] except where more details about particular IDS/IPS systems are given.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Ontology Terminology and Formal Definitions

A. Basic Terminology

Classes represent *concepts* in the domain and not the words that denote these concepts. It should be reminded that an ontology is a model of reality of the world and the concepts in the ontology must reflect this reality. Classes are the focus of most ontologies. The name of a class may change if we choose a different terminology, but the term itself represents the objective reality in the world. For example, a class of *Shrimps* can also be renamed as *Prawns* however the class still represents the same concept [4].

Slots are the properties of each concept describing various features and attributes of the concepts. It is also called a *role* or *property* [4].

Terminology is a theory of the labels of *concepts*. The labels of concepts are named after coming to an arrangement on them which involves a process of discussion in the certain *community*. The name of a class may change if a different terminology is chosen, but the term itself represents the objective reality in the world [4].

A *Taxonomy* is a hierarchy of concepts which defines relationship between concepts with the help of links such as an "is-a" or "part-of" link [7].

A *vocabulary* is a set of words where each word indicates some concepts. Vocabulary is language dependent [7].

An *axiom* is a declaratively and rigorously represented knowledge which has to be accepted without proof. In predicate logic case, a formal inference engine is implicitly assumed to exist.

Axioms have two roles as follow in ontology description:

1) To represent the meaning of concepts rigorously.

2) Within the scope of the knowledge represented declaratively, to answer the questions on the capability of the ontology and things built using the concepts in the ontology [6].

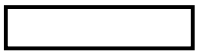
Finally a formal ontology is axiomatic description of an ontology. It can answer questions about the capability of ontology. An ontology is an explicit and less ambiguous description of concepts and relations among them appearing in the target thing. Such ontologies exist as many as the possible target things. We do not have to use logic to describe it. Formally an ontology consists of terms, their definitions and axiom relating to them; terms are typically organized in a taxonomy [6].

B. Symbols

A formal definition of the ontology for ISMO requires certain instruments such as symbols including links to slots and concepts, etc as well as axioms.

The following symbols are used for the definitions of the ontology construction. As shown below the concept/class is represented by a rectangle and the slot/attribute is shown by ellipse/oval. It can be noticed that the links between concepts, slots are represented by two different arrows indicating the *part-of* or *is-a* relationship. The first arrow shows the *part-of* relation between concept to concept and second arrow shows the *part-of* relation for concept to slot respectively.

Concepts/Class:



Slot/Attribute:

Link to (Sub)Class/Concepts:

Link to Attribute/Slot:

Table I shows the existing symbols in [8] which have been used to assist in representing the axioms designed specifically for ISMO.

TABLE I
SYMBOLS USED TO REPRESENT THE AXIOMS USED FOR ISMO

Symbol	Meaning
$\forall$	For all
$\exists$	There exist
$\neg$	Not
$\wedge$	And
$\vee$	Or
$\rightarrow$	Implication
$\leftrightarrow$	Equivalent
$\subseteq$	Belong to
$\cup$	Union

C. Axioms

Some axioms used here to design of ISMO are borrowed from [7-8].

A *part* is a *component* of the *artifact* being designed. The artifact itself is also viewed as a part. The concept of 'part' introduced here represents the physical identity of the artifact, software components and services. The structure of a part is defined in terms of the hierarchy of its components parts. The relationship between a part and its components is captured by the *predicate partOf*. Between two parts *x* and *y*, *partOf* (*x*,*y*) means that *x* is a part/components (subpart) of *y*.

The following two axioms state that a part cannot be a component of itself and it is never the case that a part is a component of another part which in turn is a component of the first part. This shows that the relation *partOf* is non-reflexive and anti-symmetric:

$$(\forall x)\neg partOf(x,x) \quad \text{(A1)}$$

$$(\forall x,y)partOf(y,x)\rightarrow\neg partOf(x,y) \quad \text{(A2)}$$

The relation *partOf* is transitive; that is, if a component of another part that is a component of a third part, then the first part is a component of the third part.

$$(\forall x,y,z)\ partOf(z,y) \wedge partOf(y,x) \rightarrow partOf(z,x) \quad \text{(A3)}$$

A part can be a (sub) component of another part. But since each part has a unique ID (its name), it cannot be sub-component of two of more distinct parts that are not components of each other.

$$(\forall x,y,z)\ partOf(x,y) \wedge partOf(x,z) \rightarrow y \leftrightarrow x \vee partOf(y,z) \vee partOf(z,y) \quad \text{(A4)}$$

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Parts are classified into two types depending upon the *partOf* relationship it has with the other parts in the hierarchy. The two types are: primitive and composite.

- A primitive part is a part that can not be further subdivided into components. These types of parts exist at the lowest level of the artifact decomposition hierarchy. Therefore, a primitive part cannot have sub-parts.
  $(\forall x)\text{primitive}(x)\rightarrow(\neg\exists y)partOf(y,x)$ (A5)
  Primitive parts serve as a connection between the design stage and the manufacturing stage.
- A composite part is a composition of one or more parts. A composition part cannot be a leaf node on the part hierarchy; thus, any part that is composite is not primitive.
  $(\forall x)\text{composite}(x)\rightarrow\neg\text{primitive}(x)$ (A6)

More composite parts are assemblies that are composed of at least two or more parts.

$(\forall x)$ assembly$(x)\leftrightarrow(\exists y,z)\ partOf(y,x)\wedge partOf(z,x)\wedge y\neq z$ (A7)

Sometimes a designer may need to find out the direct component of a part. A part is a direct component of another part if there is no middle part between the two in the product hierarchy.

$(\forall y,z)\text{direct\_}partOf(y,z)\leftrightarrow partOf(y,z)\wedge(\neg\exists x)partOf(y,x)\wedge partOf(x,z)$ (A8)

That is, *y* is a direct part of *z* if *y* is a component of *z* and there is no *x* such that *y* is a part of *x* and *x* is a part of *z*.

If *y* is a part of *x* then *x* is the whole of *y*

$(\forall x,y)partOf(y,x)\leftrightarrow wholeOf(x,y)$ (A9)

Classes are disjoint if they cannot have any instances in common:

$(\forall x,y)\text{disjoint}(x,y)\rightarrow(\neg\exists z)partOf(z,x)\wedge partOf(z,y)$ (A10)

III. Design of Identification Subsystem Modelling Ontology (ISMO)

One of the key important factors in designing a new ontology is efficiency in design. This can be achieved by splitting the ontology into several component ontologies. We call this a "collaborative design". In collaborative design each component of ontologies will be built first and then they all are compiled into a unique and unified ontology. To accomplish this, it is necessary that every component of ontologies identify separately according to their domain or conceptual level.

Therefore in order to design the ISMO, all the component of the ISMO will be designed separately and then they are all compiled and subsequently composed a single unified ISMO ontology.

A. Identification Ontology

The Intrusion Identification Subsystem (IIS) is capable of identifying attacks against a host and network (Hybrid system) by mirroring traffics and performing the following action:

- Identification: Identifies malicious attacks on host and network (Hybrid) resources.

It should also be noted that this is a temporary defense mechanism; it is not a permanent prevention of attacks like legacy IPS. The identification subsystems would use the following methods for identifying malicious attacks based on previous research [1,2] and Fig. 3.

The identification's composite for, their types, cardinality, other constraint/facets of the Identification ontology are also shown in Fig. 3, subsequently the complete identification subsystem's ontology will be represented later when all the components ontology for Identification accumulate in order to compose the identification subsystem ontology.

A value-type facet shown below describes what types of value can fill in the slot or concept. The most common value type is alphanumeric, string, number and enumerated. Some systems distinguish only between single cardinality by allowing at most one value and multiple cardinalities by allowing any number of values. For simplicity the following symbols are used to represent data/value types:

A: Alphanumeric, E: Enumerated, N: Number and S: String

The approach used in Table II is to represent the composite types and other facets, also used in [4].

TABLE II
THE CONCEPTS FOR IDENTIFICATION ONTOLOGY AND THEIR FACETS

Ontology: Identification Subsystem:				
Slot	Type	Cardinality	Other Facets	Allowed Value
Hybrid	E	Single	Class= Identification Method	Signatures
Misuse	E	Single	Class= Identification Method	Various Methods
Anomaly	E	Single	Class= Identification Method	

Similar approach to represent axioms in Table II has also been used in [9].The ontology in Fig. 2 follows the axioms as *partOf* and primitive.

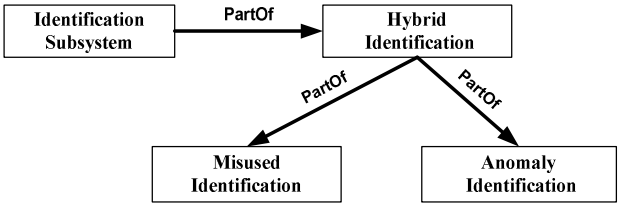


FIG. 2. BASIC VIEW OF IDENTIFICATION ONTOLOGY

For simplicity, following letters are used to represent different concepts.

A: Identification subsystem
A1: Hybrid Identification method
B: Misused Identification method
C: Anomaly Identification method

It can be noted that B and C are both subset of A1, therefore:

$B,C \subseteq A1$

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

However A1 itself is also a subset of A, hence all concepts are subset of concepts A, hence:
B, C, A1 ⊆ A
Formally it can be seen that: A=A1∪B∪C

In the Identification ontology, there must be a Hybrid, Misused and Anomaly Identification methods that can be represented by axioms 7, 8. The concepts A, A1, B and C altogether define the whole Identification ontology represented by axioms shown in Table III.

The Misused Identification method can be expressed by Rule Set Data Base (RSDB) concept. The RSDB concept belongs to the class Misused as shown in Fig. 3.

TABLE III
AXIOMS FOR IDENTIFICATION ONTOLOGY

ID	Axioms	ID	Axioms
1	partOf(Hybrid,Identification)	2	Composite(Hybrid)
3	partOf(Anomaly ,Hybrid)	4	partOf(Misuse,Hybrid)
5	Composite (Anomaly)	6	Composite (Misuse)
7	(∀x, ∃y) Identification (x) ∧ Hybrid (y) ∧ partOf(Hybrid, Identification) ∨ True		
8	(∀x, ∃y) Hybrid (x) ∧ Anomaly (y) ∧ partOf(Anomaly ,Hybrid) ∨ True		
9	(∀x, ∃y) Hybrid (x) ∧ Misuse (y) ∧ partOf(Misuse ,Hybrid) ∨ True		
10	(∀x, ∃a,b,c) Identification (x) ∧ Hybrid (a) ∧ Misuse(b) ∧ Anomaly(c) ∧ partOf(a, x) ∧ partOf(b, x) ∧ partOf(c, x)		

For simplicity notation B1 is used to represent the RSDB concept, therefore:
B1: Rule Set Data Base (RSDB)
And B1 is a subset of B, hence:
B1 ⊆ B

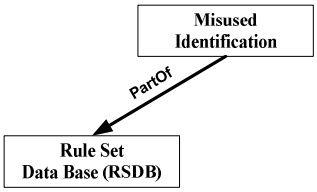


FIG. 3. MISUSED ONTOLOGY

Table IV shows the axioms that are followed by Misused ontology.

TABLE IV
AXIOMS FOR MISUSED ONTOLOGY

ID	Axioms	ID	Axioms
1	partOf(RSDB,Misuse)	2	Composite(RSDB)

The RSDB concept in turn can be expressed by the attributes/slots Methods A1, Method A2 and Method An. The different methods will be described in the next section. These attributes belong to the class RSDB as shown in Fig. 4.

The following notations are used to represent slots of RSDB concept:
b1: Method A1, b2: Method A2,….., bn: Method An
It can be noted that these slots are subsets of concept B1, that is:
b1⊆B1, b2 ⊆ B1,…., bn ⊆ B1
Hence: B1=b1∪b2…..∪bn

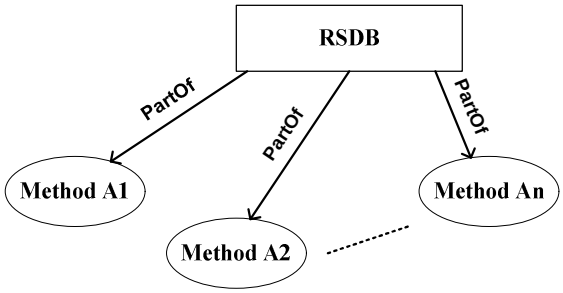


FIG. 4. RSDB CONCEPT

Table V shows the axioms abide by the RSDB concept.

TABLE V
AXIOMS FOR RSDB CONCEPT

ID	Axioms	ID	Axioms
1	partOf(MethodA1, RSDB)	2	partOf(MethodA2, RSDB)
3	partOf(MethodAn, RSDB)	4	Primitive(Method A1)
5	Primitive(Method A2)	6	Primitive(Method An)

A. Misuse Identification Methods:

In general methods for Misused identification are summarized as follow:

- ***Policy Compliance***
- ***Signature Recognition***

These are recorded signatures from previous attacks which system holds these signatures in its database. The system should recognize the intrusion by their defined markings or the vendor-specific fingerprints and identify these previously documented attacks by looking at its recorded database.

- ***Custom Signature Based on New "wi-fi" Protocols***

These are recorded signatures for new "wi-fi" Protocols. The anomaly identification is based on a profile that defines normal user activity. Therefore, an anomaly based IHS should generate alarms for previously unknown attacks, as long as the new attack deviates from normal user activity. This makes the anomaly-based IHS being capable of identifying novel attacks when they used for the first time.

The Anomaly concept can be expressed by the attributes/slots Method L1, Method L2 and Method Ln. The different methods will be describes in the next sections. These attributes belong to the class Anomaly as shown in Fig. 5.
The following notations are used to represent slots of Anomaly concept:
c1: Method L1, c2: Method L2, ..., cn: Method Ln
It can be noted that these slots are subsets of concept C, that is:
c1⊆C, c2 ⊆ C,…., cn ⊆ C
Hence: C=c1∪c2…..∪cn

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

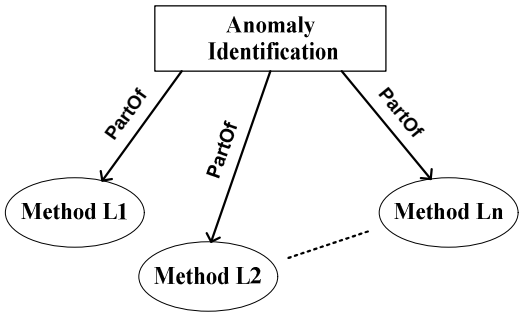


FIG. 5. ANOMALY CONCEPT

Table VI shows the axioms abide by the Anomaly concept.

TABLE VI
AXIOMS FOR ANOMALY CONCEPT

ID	Axioms	ID	Axioms
1	partOf(MethodL1, Anomaly)	**2**	partOf(MethodL2, Anomaly)
3	partOf(MethodLn, Anomaly)	**4**	Primitive (Method L1)
5	Primitive (Method L2)	**6**	Primitive (Method Ln)

B. Anomaly Identification Methods:

In general the methods used by anomaly Identification system are briefed below:

- ***Protocol Specification and Analysis (Live Packets Analysis)***
  Identification of novel attacks, risks and threats that have not been previously recorded or documented. This is mainly based on intruders approach and the way the intruder breaks 802.11 protocols of communication

- ***Policy Compliance/Script***
  Policy compliance deviation is achieved by monitoring the normal behavior of device. It is determined on a customized basis for acceptable behavior for each device. This program contains set of rules that describe what types of activities are potentially considered as a vulnerability to the system. They would monitor and analyses the network events and would initiate actions based on these analysis.

- ***Statistically Anomalous Behavior / Unusual activities***

  Dynamically alert the operator in real-time to abnormal behavior of network devices, such as repeatedly failed login access to a system or transformation of 5 MB file from one wireless host to another host at 5 am.

Some of the Misused and Anomaly methods are summarized below:

- Alerts based :
- send alerts to administrator:

email, pager, log alerts to event logs, log alerts to syslog

- Protocol based:

Filter for SYN/FIN/RST TCP packets, process TCP fragments, FTP analysis, identify and log TFTP sessions, flag HTTP-based worm sources such as Code Red, ICMP analysis, detailed analysis of http requests, detailed analysis of http replies. DNS analysis.

- Logging Based:

Detects password scans, rlogin/telnet analyser, access and record connection events.

- Others:

Real-time, detects vulnerability, port scans and incoming and outgoing connections that are ssh, record and analyse RPC portmapper requests and email traffics, track software versions, looks for blaster worms, synflood attacks, ssl analyzer, backdoors and clear text passwords.

In summary, the Identification ontology now can be composed by collaboration of Hybrid, Misused and Anomaly ontologies and can be formally described as follow:

$A=A1 \cup B \cup C$

Fig. 6 shows a final view of Identification ontology:

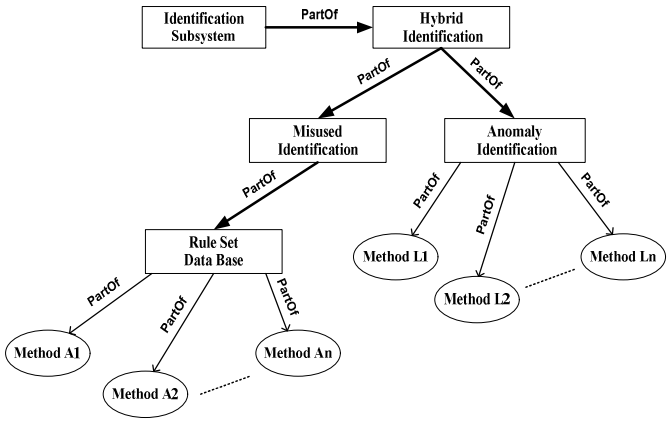


FIG. 6. FINAL VIEW OF IDENTIFICATION ONTOLOGY

IV. CONCLUSIONS

A unique IHS reference model's structure was described in this paper which employs an ontology approach to define Identification ontology modelling. A novel Identification subsystem have been designed and proposed for IHSs in WLANs. Use of such reference model should allow the characterization of different IHSs in a standardised and efficient format.

The future papers will demonstrate the final Ontology based IHS reference model which has been evaluated using existing WLANs IHS systems to prove its efficiency and accuracy in order to compare and evaluate the existing or future IHSs for WLANs.

The next paper will focus on ontology engineering approach for the response and management console subsystems of IHS reference model.

REFERENCES

[1] S. Salekzamankhani, A. Pakštas, B.Virdee, "Towards Development of a Reference Model for Intrusion Detection Systems for Wireless LANs", IEEE Globecom 2005, Workshop on Adaptive Wireless Networks, AWIN.

[2] A. Pakštas, S. Salekzamankhani, B.Virdee, "Fighting Intrusions in Wireless LANs: A Need for the Reference

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Model". Proc. 2nd IEEE and IFIP International Conference in Central Asia on the Next Generation of Mobile, Wireless and Optical Communications Networks, (ICI 2006), Tashkent, Uzbekistan , Sep. 19, 2006.

[3] T.R. Gruber, "A translation approach to portable ontology specifications", Proc. of JKAW, 1992. pp 89-108. Available: http://portal.acm.org/citation.cfm?id=173747

[4] N.F. Noy, D.L. McGuinness, "Ontology Development 101:A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford University, March 2001.
Available:http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[5] M.A. Musen, "Dimensions of knowledge sharing and reuse", Computers and Biomedical Research 25, 1992, pp 435-467.
Available: http://portal.acm.org/citation.cfm?id=176404.

[6] R. Mizoguchi, M. Ikeda, "Towards ontology engineering", Proc. Joint 1997 Pacific Asian Conference on Expert Systems/Singapore International Conference on Intelligent Systems, 1997, pp.259-266.

[7] M.A Rahman, A. Pakstas, F. Z. Wang, "Towards Communications Network Modeling Ontology for Designer and Researchers", 10th IEEE International Conference on Intelligent Engineering Systems 2006(INES2006), London, June 26-28, 2006.

[8] J. Lin, M.S. Fox, T. Bilgic, "A requirement ontology for engineering design", Concurrent Engineering: Research and Applications, Vol. 4, No4, Sept1996, pp279-291. Available:http://cer.sagepub.com/cgi/content/abstract/4/3/279.

[9] P. G. Mian, R.A. Falbo, "Building Ontologies in a Domain Oriented Software Engineering Environment", IXA Argentine Congress on Computer science(CACIC 2003), La Plata, Argentine, 6-10 October 2003,pp 930-941.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)