ONTOLOGY ADAPTER
Network Management System Interface Model

Lingli Meng, Lusheng Yan
Network Management Research Center, Beijing University of Posts and Telecommunications
Beijing, Popular Republic of China
mickey200002@sina.com, lsyan@bupt.edu.cn

Wenjing Li
Department of Computer science and technology, Beijing University of Posts and Telecommunications
Beijing, Popular Republic of China
wjli@bupt.edu.cn

Keywords: Ontology Adapter, Interface Model.

Abstract: This paper proposes a new method to define the interface model of network management system, that is ontology adapter. This model includes three parts, which are ontology agent, ontology knowledge base and ontology resource description. We can realize the uniform presentation of different network resource interface information using them. Therefore, we can take this model as a common data platform to offer the interface information to the network management system.

1 INTRODUCTION

The telecommunication network and business have such a great speed development, which cause more network resource needed to be managed in the managed network system (Luoming,2001). However, the traditional network management interface model of the TMN proposed by the ITU-T is not suitable for changes of the network technology, and can only satisfy the static management of the managed network resources, adjusting badly to all kinds of the existing interface technology. Therefore, the interface model is encountering the great challenges (Luoming ,2003)(Zhipeng,2007):

1) variety of managed network resources. For the complicated managed network resources are included under the background of network management integration, it is an important problem how to describe these managed resources in the interface layer, and give the uniform description of them before offering to the NMS(Network Management System).

2) diversity of interface description. At present, there exactly exists the diversity definition of private interfaces among all equipment vendors, although the ITU-T has standardized these interface descriptions. It can cause the inconsistence presentation of the data, and not share the managed resources with different systems, as well as be deficient in flexibility when NMS collects the information from different interface equipments. Therefore, how to find a better interface model to integrate all these different descriptions of network resources, realize the uniform data presentation, which will be a creative breakout of interface model in NMS.

So as to solve these problems above, we proposal an interface adapter model using ontology language. The rest of this paper is organized as follows: Section 2 introduces the existing method of interface definition in NMS. In section 3, we give a glimpse of ontology language, presenting the ontology language OWL. We describe the detail of the ontology interface adapter model in section 4. Finally section 5 would conclude our paper and give the future work.

2 INTERFACE ANALYSIS

ITU-T proposed five layers architecture of TMN in 1990's. The five layers are composed of business management, service management, network

management, network element management and network element, which are managed by NMS.
At present, it has been defined that several methods of the interface between NMS and managed resources, Such as SNMP, CORBA and XML. They design the interface model distinguishingly according to different prospective.

2.1 SNMP

SNMP is a standard protocol to manage the IP network. It can monitor and control the variable set,as well as monitor the two data format of equipments, SMI and MIB. SNMP adopts the model of "Management Process---Agent Process". When the Management Process sends an order to the Managed Object, the Managed Object receives the order by the Agent Process. Then the Agent process also sends back the response to the Management Process. That is, the communication between Management Process and Managed Object is not in direct, but using the Agent Process.

SMI is a method of defining the management object using SNMP. It based on object-oriented, not using encapsulation, inheritance, polymorphism. So it simplifies the management information models. It is on behalf of an instance of a class by using table to represent the data type, table, column, and row. (Jorge Vergara, 2003)

2.2 Corba

CORBA is proposed by the OMG. It is an object-oriented software development as well as application platform, with the function of portable, reusable and connection, under distributed heterogeneous circumstance.

IDL, which ITU-T use to define the TMN and UTRAD. It is an object-oriented description language. By defining of the IDL, classes are interfaces between different models. They have attributes and methods. IDL can be mapped to different languages, such as C++, JAVA, Ada, Cobol and SmallTalk.

2.3 XML

XML is given by the W3C. It is an information processing tool, which has on relevant to platform, software and hardware. XML has the feature of flexibility, easy-realization, easy-extension, as well as low cost. At present, we only use this tool to define format of the configuration files and the performance files in NMS.

3 ONTOLOGY DESCRIPTION

Ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of existence. In 1990's, ontology was brought into the field of the knowledge representation, knowledge sharing, semantic reason and so on.
In recent years, it has been great development of the applications and research of ontology, which NMS has been focusing on. There may exist the possibility of the application of ontology in network management information model (Jorge, Vergara, 2003), the relations among ontology mapping, network management operation (Wong, 2005), as well as special management information model based on ontology. However, it is still short of the research to design interface adapter model using ontology language in NMS. We will propose an ontology interface adapter model next section.

3.1 OWL

Generally speaking, we use a special ontology language to describe ontology. At present, the main ontology description languages are DAML, OIL, KIF, and OWL, among which, the OWL proposed by the W3C has the great effect.

The OWL (Web Ontology Language) is designed for processing the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than those supported by XML, RDF, and RDF Schema (RDF-S), by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

OWL has been applied widely to module in most kinds of domain ontology: W3C has used OWL in the ontology model of Web Service; Using OWL defined the analysis model, which can be adapted to test circumstance (Bodkin, 2005).

4 ONTOLOGY ADAPTER DESIGN

4.1 Ontology Adapter Model Analysis

We propose and discuss the ontology adapter model in the interface definition below. We divide ontology adapter model into three parts: ontology agent, ontology knowledgebase and ontology resource description. Figure 1 shows the model.

Now we will introduce what functions the three parts have, and how to realize the uniform description of the interface information when NE management Layer sends the network resource to the NMS. First of all, the network resource information has been defined well in the NE Management Layer.

Step1, ontology agent receives resource information, represented by the SMI, IDL or XML from the NE Management Layer. Then they will be transformed to the uniform representation using OWL. If just the

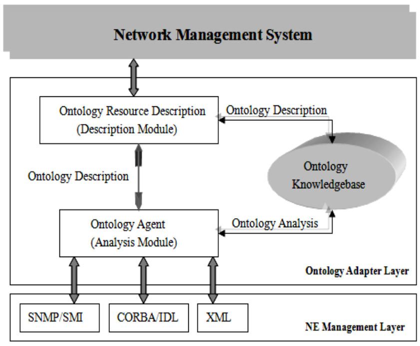

Figure 1: Ontology Adapter Model.

information describe static resources, which can be dealt with by the ontology agent; otherwise including dynamic information, which would be forwarded to the Ontology Knowledgebase. It will give a further deal.

Step2, ontology knowledgebase is used to transform the complex interface information. There exists a database, which stores more transduction rules between OWL and other interface description languages, all that the ontology agent can't deal with, will be sent to the ontology knowledgebase.

Step3, ontology resource description, in which, we can complete the interface information representation finally. After receiving the static information sent by ontology agent and the dynamic ones dealt by ontology knowledgebase, ontology resource description module will integrated both of them, and present a uniform format to NMS.

4.2 Ontology Adapter based on XML

This section we give a simple realization of ontology adapter model. We have defined some transformation rules and a simple ontology resource description frame using OWL.

4.2.1 Ontology Agent

Figure2 shows the process how ontology receives the interface information. Firstly, read interface information; Secondly, analyze language type and judge whether it present dynamic information; thirdly, transform the format, finally, the final OWL representation is formed.

In transform mode 1, take IDL as an example, there are some rules below:

Rule 1: $\forall$ module $\in$ IDL, →
xmls: $\exists$ NameSpace $\in$ OWL

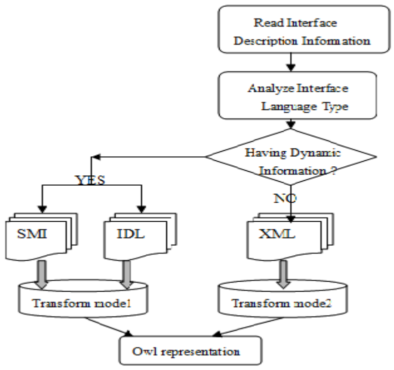

Figure 2: Ontology Agent Flow Chart.

Case study:
module TypeDemo { …… } →
xmlns="http://www.w3.org/TR/2004/REC-owl-guide-20040210/TypeDemo#"
xml:base="http://www.w3.org/TR/2004/REC-owl-guide-20040210/TypeDemo#"

Rule 2: $\forall$ Interface $\in$ IDL, → $\forall$ class $\in$ OWL

Case study:
interface Father { ……. }; →
<owl:Class rdf:ID="Father"/>

Rule3: $\forall$ Attribute $\in$ IDL, →
$\forall$ property $\in$ OWL

Case study:
attribute string name; →
<owl:DatatypeProperty rdf:ID="name">
<rdfs:domain rdf:resource="#Father"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

Role4: $\exists$ fundamental datatype $\in$ IDL, →
xsd: $\exists$ datatype $\in$ OWL

Case study: String → xsd:string

Note: there are some data types that can't be mapped directly, therefore, you must do some changes. Such as, char → xsd:byte.

Rules above, which just aim at static information. There are still methods to represent dynamic information, such as operation and context. We do such analysis in ontology knowledgebase. Simply, we use OWL text to descript these methods. Here, we only offer a sample:

e.g. Method1 (var1, var2) $\in$ IDL →
Method1 has var1, var2 $\in$ OWL.

In transform mode 2, we translate XML into OWL, we should distinguish the differences between the two kinds of data types, the fundamental data type and the user-defined ones. Because the user-defined data type is so complex that we must avoid the semantic loss at our best, while transforming such information.

4.2.2 Ontology Resource Description

In this part, we represent a uniform network resource description framework using OWL. When ontology agent and ontology knowledgebase send the transformed information to ontology resource description, it will integrate these two parts, and give a uniform representation using OWL. We now give a sample description, as follows:

The definition of class:

```
<owl:Class rdf:ID=Equipment>
  <rdfs:subClassOf rdf:resource=PhysicalME/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasDatatypeProperty rdf:resource=
      equipType>
        <owl:ConstraintRequirement
        rdf:datatype="&xsd;string">
        </owl:ConstraintRequirement>
      </owl:hasDatatypeProperty>
      <owl:hasObjectProperty
      rdf:resource=hasInterface/>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=
      "#hasInterface"/>
      <owl: minCardinality
      rdf:datatype="&xsd;int"> 1
      </owl: minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=
      "# hasInterface "/>
      <owl:haslValuesFrom rdf:resource=
      "# Interface "/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

The detail definition of attributes for the above:

```
<owl:DatatypeProperty rdf:ID=" equipType ">
    <rdf:type rdf:resource=
    "&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Equipment"/>
    <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="hasInterface">
    <rdfs:domain rdf:resource="#Equipment"/>
    <rdfs:range rdf:resource="#Interface"/>
    <owl:inverseOf
    rdf:resource="#belongEquipment "/>
</owl:ObjectProperty>
```

5 CONCLUSIONS

It is a challenging research of the integration of network resource interface information. The ontology adapter model we presented has solved how to integrate these different interface informations in architecture. Our future work will concentrate on the research of offering a specific method and realization, using which we can success to transform existing interface presentation language into OWL. If possible, a tool which can be used to produce this transformation automatically is expected.

REFERENCES

Luoming, Meng. Qi feng. *Modern Network Management Technology* ,M. Beijing: Press of BUPT, 2001.

Luoming, Meng. Network Management: Problems, Progress and Prospect, J.*Journal of BUPT*, 26-2: 1-8, 2003.

Zhipeng,Gao. Ontology-based Modeling Methods,Model and Application of Shared Management Information, D. Beijing: *Doctoral Dissertation of BUPT*, LAB. of BUPT,2007.

Jorge.E López de Vergara, Víctor A. Villagrá,Juan I. and Julio Berrocal. Ontologies: giving semantics to network management models, J. *IEEE Network*, 17-3:15-21, 2003

Wong, A.K.Y. Ray, P. Parameswaran, N. Strassner, J. Ontology mapping for the interoperability problem in network management, J. *Selected Areas in Communications, IEEE Journa*l on. 2005, Volume: 23 Issue: 10: 2058 – 2068.

Bodkin, M., Harris, M., Helton, A. Use of ontologies for decision support generation and analysis,A. *Autotestcon, 2005. IEEE.* 2005:684