Integrating Heterogeneous IT/Network Management Models using Linked Data

Kenichi Ogaki, Masanori Miyazawa and Michiaki Hayashi
KDDI R&D Laboratories
Saitama, JAPAN
{ogaki, ma-miyazawa, mc-hayashi} at kddilabs.jp

Shinichi Kadono
KDDI Corporation
Tokyo, JAPAN
sh-kadono at kddi.com

***Abstract*—Today's ICT system necessarily involves multiple system domains complying with different management models. Integration of domain-specific management models is required to achieve, for example, agile fault localization in a huge ICT system, but this has not been efficiently achieved due to the difficulty in bridging multiple management models. To address this issue, we propose a systems management architecture reusing an existing standardized system management model for each domain and integrating them by introducing meta-data modeling. To show the feasibility of our proposal, we demonstrate that managed data of IT and network resources modeled by Common Information Model (CIM) and Shared Information Data (SID) model are bridged by meta-data modeling based on Resource Description Framework (RDF). The architectural design implies reduced effort thanks to use of well-deployed and standardized models and meta-data modeling language with its easy-to-use query language.**

Keywords—SID, CIM, Linked Data, RDF, SPARQL

I. INTRODUCTION

Today's ICT system involves not only network resources but also IT resources, such as servers and storages. Hence, the operator requires the integration of multiple domain-specific management models in order to achieve the unified management of system faults, for example. However, an integration has always caused problems because the managed data of each domain are based on a different standard model. When we look at today's telecommunications carriers, they are engaged in Internet services which require not only network resources but also huge IT resources in their own datacenters. Thus, the inventory repository of resources is deployed for each domain-specific system, and this situation has been raising development costs of operating support systems (OSSs) tailored to each repository. As a simple example of a failure effect analysis across IT and network domains, an OSS must traverse both repositories to identify affected services when a failure occurs in a part of the network. In terms of systems management models that define the semantics or ontologies of ICT resources and services, they are standardized in different standardization organizations (SDOs). In IT and network resource models, CIM [1] and SID model [2] are de facto standards and have been supported.

Thus, the biggest challenge is to integrate multiple repositories with different domain-specific standardized management models in order for operators to easily operate a large ICT system. Our research goal is not to create a unified model and enforce it on different domains, but to design an architecture to easily handle any data from multiple repositories across different domains. To achieve this, we propose to introduce a common meta-data modeling language to describe different standardized systems management models in the same manner. Then we chose a language from evolving semantic web and linked data technologies. This paper employs the best practice of utilizing semantic web and linked data technologies to tackle connecting larger data from heterogeneous systems. To prove the concept, we demonstrated the feasibility by using system analysis applications across IT and network systems domains. These resource management models (i.e., CIM and SID model) and the resource data themselves were handled as RDF data [3]. The architectural advantage of our approach is the easy integration of a new repository with little effort because the managed data following an existing standardized model are just expressed as RDF data without redefining a new model and mitigated coordination between different models is required only at the boundary between two domains. Furthermore, an easy-to-use standardized query language SPARQL [4] like SQL shall reduce efforts to develop a system management application. This approach also improves the performance of traversal queries of linked data for various system analyses.

The rest of the paper is organized as follows. Section 2 reviews related work and presents the approach. Sections 3 discuss our proposed architecture. Section 4 reports the results of some demonstrations. And Section 5 provides concluding remarks.

II. RELATED WORKS

A typical approach to integrate the management of different domain-specific systems is to model these domains using a common meta-model like UML. Sethi et al. presented a framework for defining the model of a newly introduced component using UML and relating it with models of other existing components through the semantic relationship remaining in their models isolated from each other, which resulted in federating various data sources [5]. This framework, however, requires to normalize and define a new management model. To eliminate this modeling cost, a well-defined system management model by SDO for each domain should be reused. Thus, in our previous work, we adopted UML-based SID model. However, we still needed to extend this model to support IT resources by ourselves [6]. In the semantic web and linked data technologies area, connecting huge data on the

978-3-901882-50-0

web is being tackled so as to be semantically utilized, and World Wide Web Consortium (W3C) standardized a common meta-data modeling language, RDF, and a ontology language, Web Ontology Language (OWL) [7], to express linked data and handle these. Feridun and Tanner presented a loosely coupled integration of data to be easily browsed, searched, and queried across multiple sources based on semantic web and linked data technologies without enforcing a common model, and demonstrated the implementation of linked data interfaces for their own products [8]. They, however, just implemented a kind of one-to-one translator to handle data from their own management systems using RDF, and did not consider an overall systems management architecture which integrated multiple different well-deployed and standardized models.

In our aforementioned previous work [6], we also adopted a proprietary object-oriented database system to efficiently manage resource data defined by an object-oriented model (i.e., SID model). Many domain-specific resource models are today formulated as object-oriented models. Thus, to efficiently handle relationships among these managed data, a graph- or object-oriented database, which is optimized to handle graph-structured data, is desired. On the other hand, to reduce system development and operating costs, an adoption of a proprietary database architecture should be avoided and a standardized one should instead be used.

From reviewing these works, unaddressed challenges to integrate heterogeneous systems managements are (1) to reuse the well-deployed and standardized management model for each domain-specific system without redefining a new unified model, (2) to express these multiple different management models by using a common meta-data modeling language so as to handle managed data across domains and (3) to show the feasibility, scalability and performance for common system management applications based on real-life heterogeneous systems, by storing data into a standardized database which fits existing object-oriented systems management models. Furthermore, as explained at the outset, the management integration of IT and network systems especially interests us, telecoms carriers, and the inconsistencies between these management models are anticipated. Thus, to employ the best practice in addressing these challenges in this paper, we integrated and handled IT and network resources defined by CIM and SID model, as RDF data, and demonstrated a fault analysis application across the IT and network system domains.

III. Integrating Heterogeneous Systems Management Models using Linked Data

We adopted RDF to describe resource entities defined by each model in the same manner and stored their data into repositories capable of handling RDF. RDF expresses a resource entity as a set of triples: each triple is a simple data structure composed of subject, predicate and object. A predicate represents the relationship between the subject and the object, like the subject connecting to the object or the subject having the array of an object. An object represents a value, like a string or numerical value, which elaborates the subject, or a link to the other subject expressed as Uniform Resource Identifier (URI). From a stored set of triples, an arbitrary attribute of an entity can be looked up by using the easy-to-use query language SPARQL. The data can be easily queried based on the semantics of the resource relationships, such as a traversal query on graph-structured data, and can be CRUD(Create/Read/Update/Delete).

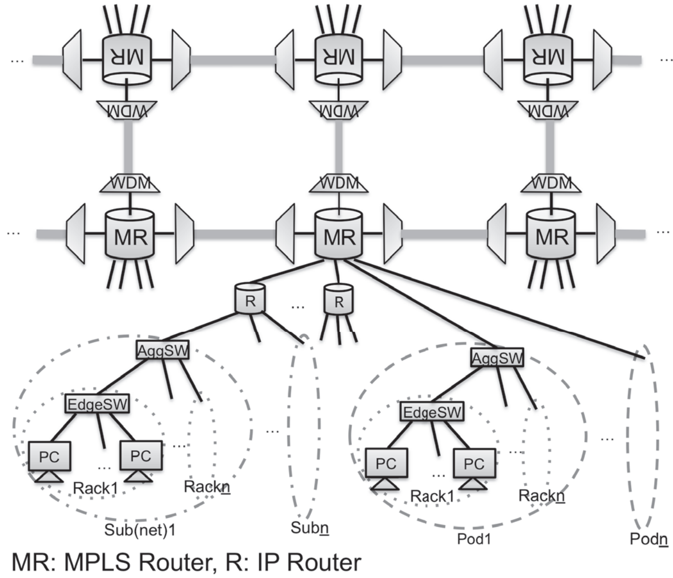

Fig. 1. An example of IT and network system

The following describes a modeling example through integrating IT and network systems management in a current carrier's operation with this architecture.

A. Heterogeneous Systems Management Modeling

1) Network System Management Modeling

Historically, telecoms carriers' operation architecture has been standardized at TeleManagement Forum (TMF). Many carriers have been operating their network based on this architecture and OSS vendors' products support TMF SID model for managing network resources. In the network part of the IT and network system shown in Fig. 1, the logical relationships among edge and aggregation Ethernet switches (EdgeSW and AggSW), IP and MPLS routers (R and MR), and WDM equipment can be modeled by using a subset of SID

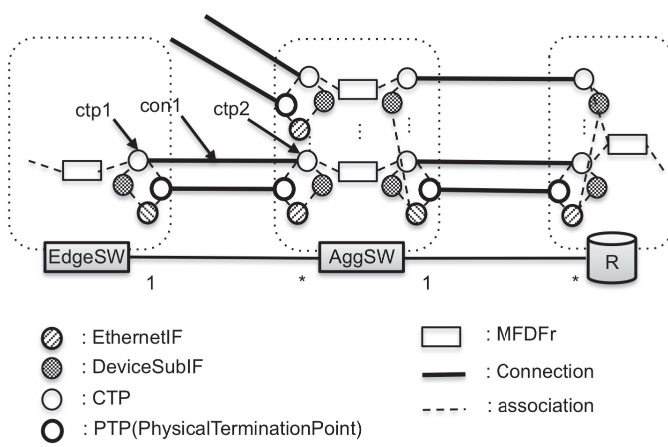

Fig. 2. A simple network modeled by SID model

resource domain model.

As an example of focusing on the relationship from a edge switch to a router through an aggregation switch, this part of the network can be modeled as instances of SID model classes, such as Connection Termination Point (CTP) class, which terminates a logical connection, MatrixFlowDomainFragment (MFDFr) class represents a packet switch, and Connection class represents a connection between two facing CTPs as shown in Fig. 2. All network resources can be modeled as such class instances and an instance can then be expressed as multiple RDF triples as aforementioned. For example, the relationship between the edge switch and aggregation switch shown in Fig. 2 is described as follows:

subject	predicate	object	
ctp1	_pipe	con1	(1)
con1	_terminationPoint	ctp1	(2)
con1	_terminationPoint	ctp2	(3)

(1) Instance ctp1 of the CTP class has instance con1 of the Connection class as the value of a member _pipe, meaning that ctp1 terminates the connection con1. (2) con1 has ctp1 as the value of a member _terminationPoint, meaning that one end of con1 is ctp1. (3) con1 similarly has ctp2 as the value of the member _terminationPoint. Where ctp1 is the logical termination point in the edge switch, ctp2 is the one in the aggregation switch and con1 is a connection between them.

2) *IT Systems Management Modeling*

On the other hand, the operation architecture of IT systems have been well-standardized as ITIL (Information Technology Infrastructure Library) at itSMF (IT Service Management Forum). DMTF (Distributed Management Task Force) CIM has been standardized to implement CMDB (Configuration Management DataBase), which plays a core roll in the configuration management process in ITIL, so as to define resource models in the IT environment. The relationships among entities inside a PC, i.e., the server, are modeled consisting of class instances of VirtualComputerSystem, i.e., virtual machine, OS (Operating System) and UnixProcess running on ComputerSystem, and ProtocolEndpoint that terminates a logical connection toward a network based on a subset of CIM Schema. These instances are also expressed as RDF triples as similar as network resources.

3) *Systems Boundary Modeling*

In this architecture, each different resource model is used in each different domain, and this then causes model inconsistencies at the boundary between two domains. For example, in SID model, a logical connection is terminated at the instance of the CTP class. On the other hand, in CIM, a logical connection is terminated at the instance of the ProtocolEndPoint class. Thus, in modeling a connection between an edge switch and a PC, a ProtocolEndPoint class instance is required to terminate the connection instead of a CTP class instance, or vice versa. In this case, usual RDBs or programming languages may not allow this due to the inconsistent class type. On an RDF store which stores and manages RDF data, however, a subject and an object can be easily linked with a loose coupling, meaning that a class type of the subject or object is not checked. Even though this approach works well, to improve the reusability of stored data, a consistent definition should be provided as a kind of schema. Thus, the ProtocolEndPoint class may multiply inherit from the SID Termination Point class as well as its own original parent class, or the SID Connection class may be extended to allow a ProtocolEndPoint class instance as the value of the member _terminationPoint. Furthermore, with an RDF expression of an OWL equivalentClass predicate, we can let two different classes in different models be considered as a same class.

B. *System Analysis Application by using Linked Data*

As aforementioned, when all entities and relationships among them are stored onto an RDF store, any arbitrary analysis using these data is easily realized by using the query language SPARQL. As a simple example, to find the opposite CTP of the instance ctp1 via the logical connection shown in Fig. 2, i.e., ctp2, the query expressions are described as follows:

```
SELECT ?ctp                                           (4)
WHERE{
  ctp1          _pipe               ?connection .     (5)
  ?connection   _terminationPoint   ?ctp .            (6)
  FILTER(?ctp != ctp1)                                (7)
}
```

(4) The SELECT sentence returns the variable *?ctp*, which satisfies the conditions described as RDF triples inside the WHERE clause. (5) Where the variable *?connection* returns all values of a member _pipe of ctp1, (6) the variable *?ctp* returns all values of a member _terminationPoint of all *?connection*, and (7) *?ctp* must not be ctp1.

For a feasibility study of our proposed architecture, this fault analysis example is demonstrated in the next section.

IV. Proof-of-Concept Implementation and Demonstration of Feasibility

To demonstrate our proposed architecture, we conducted a proof-of-concept implementation of the resource management integration of IT and network systems. To show the applicability to system analyses, we evaluated the performance of a failure effect analysis, as well as a comparison with an existing RDB, as a reference.

A. *Proof-of-Concept implementation for IT and Network Systems Modeling*

We modeled IT and network systems based on existing SID model and CIM, and loosely interconnected these two models. And pseudo resource data of the system shown in Fig. 1 were created and stored into an RDF store and an RDB.

As for the network system part in the right side of Fig.1, we assumed N aggregation Ethernet switches under each MPLS router, N edge Ethernet switches and N PC servers under each aggregation switch and edge switch, respectively. Then, the network resources were modeled based on the subset of SID resource domain model and the resource data were generated as RDF/N3 format by using the Sesame library ver. 2.6.3 [9]. In modeling an upstream from an MPLS router, trails that represent MPLS Label-Switched Paths (LSPs) were configured from each MPLS router to five other MPLS routers. In this demonstration, this trail contained one IEEE 802.1ad S-tagged connection and the S-tagged connection contained one 802.1Q C-tagged connection. This C-tagged connection was connected to one edge switch under each aggregation switch through MFDFr, which bridged the same Tag, and N PC servers under this edge switch. A trail was also associated with the connections between two adjacent MPLS router and WDM equipment, forming a hierarchical structure. For the IT system part, CIM data were actually collected from a Linux box and were also expressed as RDF/N3 format as the resource data of a PC server. At the boundary between IT and network systems, the CIM ProtocolEndPoint class was simply extended to be able to have _pipe as a member, which referred to an instance of the SID Connection class so as to easily interconnect these two models with a minimum coordination.

These resource data were stored into an RDF store, Virtuoso Open-Source Edition ver. 6.1.2 [10], and an RDB, PostgreSQL ver. 8.4.9 on the same Linux box. To store RDF data into PostgreSQL, the Sesame library is also used to automatically create master tables for subjects and objects and associated tables for predicates. In executing a query, Virtuoso accepted native SPARQL query expressions. On the other hand, for PostgreSQL, Sesame library automatically translated a SPARQL query to an SQL query.

B. Applicability to Failure Effect Analysis

As the feasibility demonstration of end-to-end system analysis, a failure effect analysis was performed. In simulating a failure at a logical connection between an edge switch and an aggregation switch in Pod1 shown in Fig. 1, a query for all PC servers, i.e., CIM ComputerSystem, connected to the same VLAN as this failed connection across MPLS routers, edge switches, and aggregation switches was easily described as a SPARQL query. Then, the query response time against the network scale was measured on the RDF store and the RDB (Fig. 3). The aforementioned N of N-ary tree structures of MPLS routers, aggregation switches, edge switches and PC servers was changed to 2, 4, 6, 8, and 10, which meant that the total number of PC servers in the system, $6N^3$, was changed to 48, 384, 1296, 3072, and 6000, and the number of localized PC servers by the query, $6N^2$, was changed to 24, 96, 216, 384, and 600. As shown in Fig. 3, the query response times were measured with five trials for each N on both the RDB and the RDF store, all the values were plotted overlaid and the times quadratically increased with the growth of the system scale. The fitting curves by quadratic regression analysis and the approximate equations with the R-square values of greater than 0.99 were also shown, and the quadratic coefficient of the approximate equation for the RDB was about 25 times larger than that for the RDF store. The reason of this difference is that a triple sentence in the example of WHERE clause shown in section III. B is basically translated into a JOIN operation among subject, predicate, and object tables on the RDB, and a traversal search query is then translated into a multiply nested JOIN. And RDB cannot generally perform this operation well. From this result, different resource management models of IT and network systems could be easily integrated, and end-to-end

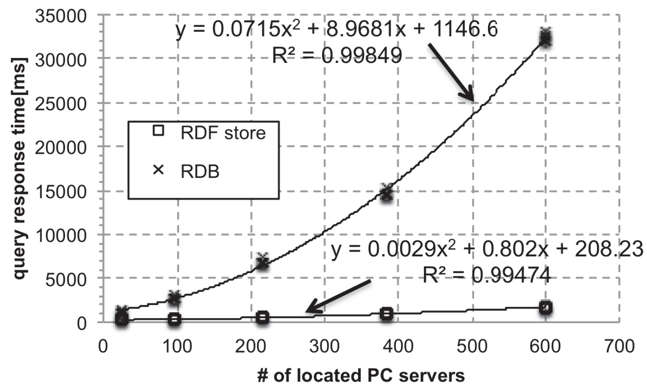

Fig. 3. A failure effect analysis performance comparison between the RDF store and RDB

system analysis across domains were demonstrated to show the feasibility of our proposed architecture. By introducing linked data technology, it was discovered that a resource model defined as an object-oriented model should be handled by the RDF store rather than RDB in system analyses with a traversal search.

V. CONCLUSIONS

Today's large system is composed of a complex combination of multiple domain-specific systems and the integration of these systems management models has been always problematic. To address this issue, we proposed a systems management architecture reusing an existing standardized systems management model for each domain and integrating them at the level of meta-data modeling language. Based on this architecture, IT resource and network resource data modeled by SID model and CIM were easily integrated and handled using RDF, and we demonstrated the feasibility of agile system analysis across these domains.

REFERENCES

[1] Distributed Management Task Force(DMTF), Common Information Model (CIM), http://dmtf.org/standards/cim

[2] TeleManagement Forum (TMF), GB922 Information Framework (SID) Solution Suite, http://www.tmforum.org/InformationFramework/1684/home.html

[3] G. Klyne and J. J. Carrol, Eds., Resource Description Framework (RDF): Concepts and Abstract Syntax, World Wide Web Consotium(W3C), http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[4] S. Harris and A. Seacorne, Eds., SPARQL 1.1 Query Language, W3C working draft, http://www.w3.org/TR/sparql11-query/

[5] M. Sethi et al., "An Open Framework for Federating Integrated Management Model of Distributed IT Environment," Proc. IEEE/IFIP NOMS 2008, pp. 803-806.

[6] M. Miyazawa and M. Hayashi, "Hierarchical Resource Model-driven Performance Analysis with Dynamic Data Control," Proc. IEEE/IFIP NOMS 2012, pp. 810-822.

[7] D. L. McGuiness and F. V. Harmelan, Eds., OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/

[8] M. Feridun and A. Tanner, "Using Linked Data for Systems Management," Proc. IEEE/IFIP NOMS 2010, pp. 926-806.

[9] OpenRDF.org, Sesame, http://www.openrdf.org/

[10] Openlink software, Inc., Virtuoso Open-Source Edition, http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/