Using Linked Data for Systems Management

Metin Feridun, Axel Tanner

Abstract—**Integration of data from multiple sources makes it possible to build effective systems management solutions. Despite the expected benefits, data integration remains a challenge. Heterogeneity between data sources in terms of lack of accepted common model, data semantics and access methods are among the difficulties. The goal of our research is to realize loosely coupled integration of data for systems management by building a lightweight mechanism to easily browse, search and query data across multiple sources without enforcing a common model across all sources. The approach is based on the emerging Semantic Web and Linked Data technologies proposed for the World Wide Web (WWW). The focus of this short paper is to report on our work on the transformation of management data sources into Linked Data providers.**

Index Terms—**Systems Management Data Integration, Linked Data, Semantic Web**

I. INTRODUCTION

Integration of data in systems management is a persistent and recurring issue. The management of any infrastructure, especially large data centers, benefits from the combination of data from multiple sources including management software systems (MSSs), custom software, and even WWW sources such as forums and self-help sites. As an example, consider that a *monitoring application* receives an alert indicating that a server has failed. Matching the server name against the *application inventory database* determines the business applications affected, allowing a priority to be assigned to the problem. Next, a specialist who can fix the problem is found from a *personnel skills database*, and her availability is checked using a *calendar* application. Finally, the actual work is scheduled and tracked using *trouble-ticket software*. There are many similar scenarios where diverse data originating from multiple systems can and need to be utilized for effective systems management solutions.

Despite its known benefits in systems management, data integration remains a challenge. Integration of components, whether software originating from the same or different vendors, legacy or proprietary, is faced with the problems of the availability of a common, high-level model to be used across resources; linkages between the different models; differences in how data is accessed, and finally, inconsistencies in the data for the same resource. Standards can help, but existing standards, such as CIM [1] and MIBs [2] for common data modeling, and WS-Man [3] and SNMP [4] for access protocols, are not widely accepted or implemented: custom solutions are generally the rule and not the exception.

The objective of our research is to realize loosely coupled integration of data for systems management. Rather than enforcing a common model across each of the many data sources, the goal is to build a lightweight mechanism to easily browse, search and query data across multiple sources, including those that are traditionally not part of systems management. The approach is based on the emerging Semantic Web and Linked Data technologies [5][6][7] proposed for the WWW, where a similar integration challenge exists at a much larger scale. The focus of this short paper is to report on our work on the transformation of management data sources into Linked Data providers as core building blocks for achieving this objective.

The paper begins with a description of the concepts of Linked Data, followed by how the latter fits into systems management. Section IV then explains the steps in creating Linked Data providers for systems management, and Section V describes a prototype built with such providers. Finally, the paper concludes with a discussion of open issues and future work to address them.

II. WHAT IS LINKED DATA?

According to the vision of the *Semantic Web* [5], entities of interest are named by and referred to using a Uniform Resource Identifier (URI). Data about an entity is expressed as a set of simple subject-predicate-object triples encoded in the *Resource Description Framework* (RDF) [8]. *Linked Data* is a term describing a set of best practices to facilitate the publishing, accessing and interlinking of the data of the Semantic Web. In addition to using URIs as identifiers for entities, Linked Data is based on the following "rules" [7]: (i) entity URIs should be dereferenceable via HTTP; (ii) upon accessing a Linked Data URI, the Linked Data provider should respond with 'useful information' about the entity in the form of RDF triples, and (iii) links to additional data URIs should be included in the response as much as possible, in order to maximize the interlinking of the web of data. These rules are referred to as the *Linked Data Principles*, and many data sources are already available based on them, including domain-specific information such as geographic and publication data, but also general information such as the Wikipedia data (as DBpedia, see [9] for more sources). The overall goal is to make any kind of structured data widely accessible via the standard HTTP protocol using a structured and machine-readable format (RDF) and supporting easy interlinking of different data sources. Linked Data is expected to provide the same type of conditions to the "web of data" as

Manuscript received September 1, 2009.

M. Feridun and A. Tanner are with IBM Research – Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland. (phone: +41 44 724 8649; e-mail: {fer|axs}@zurich.ibm.com).

those that spurred the explosive growth of the original WWW.

III. Linked Data in systems management

Thus, Linked Data is an open framework for the loose integration of data in the Internet, where data sources can easily cross-link. This flexibility is very useful in the systems management domain, where many data sources with related or overlapping data exist. Loose-coupling permits development of “good enough” solutions, in which the depth of integration can vary and, as with the WWW, the data space may occasionally contain misleading or dangling links, links to irrelevant models, etc. [10].

In contrast to other Linked Data providers, however, systems management data should be available only to a limited audience, e.g., system operators. Access control is therefore an important concern in Linked-Data-based systems management solutions and needs to be enforced across three layers: access to the overall data space, access to the Linked Data providers, and access to the actual data sources such as the MSSs. The first two can be provided through access controls at the application and Linked Data provider level, and are concerned with “read-only” data. For the third, MSS level access control needs to be used to prevent unauthorized access to available tools.

As mentioned earlier, there is no common model that is sufficient to encompass all other models and mapping between models is not trivial because of differences in syntax and semantics [11][12]. The approach here is to leave the original, MSS-specific models unchanged as much as possible when creating Linked Data providers. The main advantage of this is that new data sources can be incorporated into the data space quickly and with a relatively small effort.

Linked Data is based on the general “pull” model of the web for data gathering. Systems management applications that need to be informed of changes in data, e.g., event monitoring, will have to deploy appropriate mechanisms to handle updates to remedy this short-coming.

IV. How to create Linked Data interfaces

The implementation of a Linked Data interface for a particular MSS requires a number of design steps. These are detailed in the following subsections.

A. *Selection of data*

An MSS needs to expose only a subset of its data through its Linked Data interface. This subset should provide sufficient information to enable “good enough” management and facilitate the integration between different providers. For example, the chassis and interface cards (IP, cards, and ports) for a network switch should be exposed, but not necessarily specifics such as fans and sensors. The trade-off here is one between the level of depth of exposed information and the implementation (and update) effort needed for the Linked Data interface.

B. *Model Normalization*

Another consideration is how far the MSS internal data representation should be normalized to an external, higher-level representation. For example, if resources of different types managed by an MSS are stored in a single, large database table, the extent to which the Linked Data interface should present these as generic “entities” versus separate types, such as computer systems or network components, needs to be determined.

C. *Definition of URIs*

Given the selected MSS structures to expose, specific URIs must be defined as the representation of the entities of interest. Following Semantic Web principles, and also for practical reasons, these URIs must be unique and remain stable over time. To fulfill this, a URI needs components describing the local namespace environment of the MSS, the type and instance of the MSS, and details of the entity selected, for example like in:

`http://fusio.ibm.com/ITM/c3/CompSys/xyz.ibm.com`

Here, `fusio.ibm.com` represents the namespace environment of the MSS; `ITM` (as acronym for IBM® Tivoli® Monitoring) the type of the MSS; `c3` is the identifier of the specific ITM server instance; and `CompSys/xyz.ibm.com` represents the entity of interest, i.e., here a computer system monitored by the MSS and identified by its fully qualified name.

According to the Linked Data principles such URIs should be resolvable, and therefore the local namespace part of the URI needs to lead to an actual HTTP server. To satisfy the stable naming requirement and at the same time allow for flexible implementation, i.e., the ability to change MSS host locations, often a redirecting HTTP server is deployed as an intermediate.

D. *Model considerations*

The previous steps can, but do not have to, go hand in hand with the use of an existing, or a developed, rough ontology model for the MSS. Such a model can in turn be linked to a common, high-level model using concepts such as “related to” or “similar to” to help categorize data coming from multiple sources into loose categories. Such a model only needs to have sufficient detail to allow applications to identify groups of related data, but can be refined over time as needed.

E. *Interlinking of data*

To ensure that the data source is linked to other data sources and that its data content can easily be exploited by applications, a number of steps need to be taken:

- Links can be added to refer back to views and tools available in the source MSS for a given resource. Such links allow easy and focused access to the requested data and capabilities in the context of the source MSS, thereby avoiding duplication of data and functionality.
- New concepts and data refinements that bring additional value can be added by combining MSS data, e.g., the concept of *BGP peer routers* can be derived from lower-level BGP configuration information.
- As prescribed in the Linked Data specification, links to other known, related (external) data sources are added.

Establishing cross-links at the level of resource instances is challenging as in general, the naming of resources is not normalized across MSSs. For example, rules may be insufficient to relate the identifying information of one MSS (e.g., the fully qualified domain name of a computer) to that of another MSS (e.g., the MAC address). Another inevitable issue with cross-links is that those created by rules locally in an MSS can occasionally become "dangling" because the link target derived by the rule may not exist in the given environment.

An alternate mechanism for cross-linking data is to use "directories" (also accessible as Linked Data) that explicitly point to URIs with related information. The URIs are learned by crawling the available Linked Data providers and combining their information. Yet another mechanism is the use of available MSS ontologies to establish links between MSSs at the class level, deploying a common higher-level bridging ontology, e.g., by stating that the "entity" type of one MSS corresponds to the "host" class of another MSS.

We find that one of the advantages and strengths of using the Semantic Web standards is that there are ways to implement these kinds of data enhancement through standards-based rule inferencing or reasoning.

F. Accessibility through crawling

Support for crawling via the Linked Data interface enhances the visibility of the MSS data, and can be exploited by applications in understanding or discovering the metadata of a Linked Data provider. From a small set of starting URIs, all available information can be reached by recursively following links. An attractive option is to use the hierarchy inherent in the URI for crawling. For example, on the request for the URI

`http://fusio.ibm.com/ITM/c3,`

information containing all entity types, and on the request for

`http://fusio.ibm.com/ITM/c3/CompSys,`

information containing all computer systems known to the MSS should be returned, in both cases as RDF.

G. Linked Data provider architecture

Figure 1 shows the basic architecture of a Linked Data interface. As a Linked Data provider is accessed via the standard HTTP protocol, the entry point is an *HTTP Handler*. The incoming URI request is parsed in the *URI Parser* and translated into one or more suitable requests by the *MSS Access Logic* to be sent through the corresponding *MSS API* to the MSS. The *RDF Creator* component converts the response of the MSS into an RDF graph following the previously defined model. Using inferencing with suitable rules, the RDF graph can then be enriched with additional derived triples as described above. The resulting RDF graph is sent back, encoded as XML, as response to the URI request received by the HTTP Handler. All of these components should minimize the overhead introduced on top of the MSS API calls.

The required MSS access logic will be based on MSS-specific APIs or be generic. For example, code to access a personnel directory via LDAP can be written to be generic, and reused for other LDAP sources with some reconfiguration.

Protection of the sensitive data is identical to the quite well understood access control issue for web resources. Protocols such as HTTPS can be used to protect the transport level and supplemented with various authentication methods to implement access control together with a suitable authorization engine. As the Linked Data interface is only a front-end to an MSS, its authentication/authorization implementation will need to integrate with that of the MSS, which will require additional configuration and may be challenging, for example, if single sign-on capability is not available.

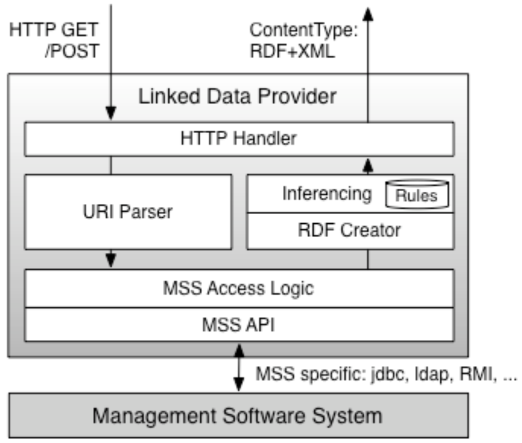


Figure 1 Components of a Linked Data provider for a Management Software System

V. Example Implementation

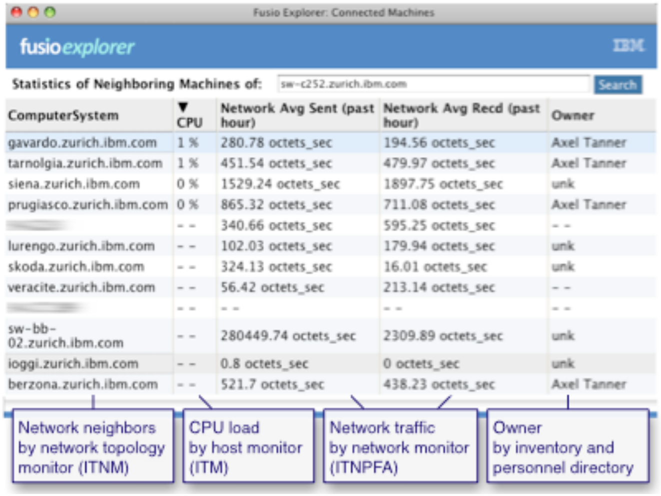
We have implemented a set of Linked Data interfaces, including interfaces for host and network monitoring systems, an inventory system and a personnel directory.

The Linked Data interfaces developed range from "specific" to "generic": The interfaces for the IBM Tivoli Monitoring (ITM) and IBM Tivoli Netcool Performance Flow Analyzer (ITNPFA) use the respective web-services APIs. The inventory system data is accessed using an exported spreadsheet file using a custom parser. On the other hand, the Linked Data interface to IBM Tivoli Network Manager (ITNM) uses generic translation from its underlying database to RDF with the help of the D2RQ library [13], and the personnel directory is accessed using a home-made, but in principle generic, translation from LDAP to RDF.

Depending on the source data, the modeling efforts differ widely. For example, data from the LDAP-accessible personnel directory can be naturally represented as RDF triples attached to a URI representing the employee. On the other hand, for an MSS with very detailed, large amount of data, identification of entities of interest and the definition of rules to infer higher-level properties from the data, e.g., from relationships between database tables, can require a significant effort. We have also noticed that inferencing with rules on top of a large database exposed completely as RDF, e.g., via D2RQ, will be computationally very intensive. This leads to a necessary trade-off between pre-running ("forward chaining") rules versus running rules on demand ("backward chaining") and the timeliness of picking up changes to the underlying database store.

As a demonstrator, we built a small special purpose, web-based application which, given the name of a computer system or network device, shows the devices directly connected to it and selected statistics (CPU load, network flows, ownership). This involves getting the network neighbors from a network topology manager, and for each of the devices identified to query host monitoring (for CPU load), network performance monitoring (for current network flows) and inventory and personnel databases (for ownership information and resolving of the full name).

A screenshot of this application is shown in Figure 2, where the query for a router *sw-c252.zurich.ibm.com* is shown, resulting in a table of connected resources with selected statistics coming from different MSSs. It can be seen that for some resources not all values are available because the respective MSS does not manage this particular resource.



Statistics of Neighboring Machines of: sw-c252.zurich.ibm.com

ComputerSystem	▼ CPU	Network Avg Sent (past hour)	Network Avg Recd (past hour)	Owner
gavardo.zurich.ibm.com	1 %	280.78 octets_sec	194.56 octets_sec	Axel Tanner
tarnolgia.zurich.ibm.com	1 %	451.54 octets_sec	479.97 octets_sec	Axel Tanner
siena.zurich.ibm.com	0 %	1529.24 octets_sec	1897.75 octets_sec	unk
prugiasco.zurich.ibm.com	0 %	865.32 octets_sec	711.08 octets_sec	Axel Tanner
[redacted]	- -	340.66 octets_sec	595.25 octets_sec	- -
lurengo.zurich.ibm.com	- -	102.03 octets_sec	179.94 octets_sec	unk
skoda.zurich.ibm.com	- -	324.13 octets_sec	16.01 octets_sec	unk
veracite.zurich.ibm.com	- -	56.42 octets_sec	213.14 octets_sec	- -
[redacted]	- -	- -	- -	- -
sw-bb-02.zurich.ibm.com	- -	280449.74 octets_sec	2309.89 octets_sec	unk
ioggi.zurich.ibm.com	- -	0.8 octets_sec	0 octets_sec	unk
berzona.zurich.ibm.com	- -	521.7 octets_sec	438.23 octets_sec	Axel Tanner

Network neighbors by network topology monitor (ITNM)
CPU load by host monitor (ITM)
Network traffic by network monitor (ITNPFA)
Owner by inventory and personnel directory

Figure 2 Screenshot of demonstrator application

All components (application and Linked Data interfaces) are implemented as Java^{TM} servlets running in a web application server (Apache Tomcat). RDF-related components use the Jena library [14].

VI. Experience and Future Work

Use of Linked Data in systems management is promising as it provides a platform where data can be loosely integrated and further refined within the familiar environment of the World Wide Web. Our experience with the prototype we built demonstrates that unifying data from multiple sources for specific queries is relatively easy to construct, without having to duplicate or undertake major transformations of source data.

The next step in our research is to address some of the challenges we identified during our work. First is the simplification of the design of Linked Data interface, i.e., a tool that helps cross-link data by identifying relationships between available data models. This tool will create and maintain a catalog of data concepts collected from the available data sources. It will also actively search data sources to discover syntactically similar or related data concepts and instances. Linked Data interface designers will use the tool to identify the links that should be added to the Linked Data model without the necessity to know other (linked to) data models in depth.

Complex, distributed queries over available Linked Data providers are required, but difficult to provide. A search index can be used as de facto router for simple search queries. A common, high-level model of classes, e.g., “ComputerSystem” to which a data source can link to, could allow limited vocabulary and distributed searches. Distributed, complex queries similar to database queries are more difficult to achieve. Semantic Web Client Library [15] is an implementation of federated query resolution in which URIs from Linked Data providers are iteratively fetched and subjected to the given query. The challenge is to extend such approaches to systems management, where specialized models limit the number of links between data and the amount of data extracted during query processing needs to be restricted to avoid performance problems.

The inclusion of sources of unstructured data such as search engines, forums, and free-form logs as Linked Data providers is an interesting problem. A Linked Data provider will require source-specific analytic tools to extract the necessary fields from the data. Challenges here are analysis, i.e., what fields to extract, scalability and the timeliness of content.

References

[1] Common Information Model (CIM), a DMTF standard [Online]. Available: http://www.dmtf.org/standards/cim/

[2] K. McCloghrie, D. Perkins, and J. Schönwälder. Structure of Management Information Version 2 (SMIv2). RFC 2578, April 1999.

[3] Web Services for Management (WS-Man), a DMTF standard [Online]. Available: http://www.dmtf.org/standards/wsman/

[4] Simple Network Management Protocol (SNMP), Internet RFC [Online]. Available: http://tools.ietf.org/html/rfc3411

[5] T. Berners-Lee, J. Hendler and O. Lassila, “The Semantic Web,” Scientific American, Vol. 284, No. 5, pp. 34-43; May 2001

[6] Tim Berners-Lee: “Linked Data” [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html

[7] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - the story so far," Int’l J. on Semantic Web and Information Systems (IJSWIS), 2009

[8] Resource Description Framework, a W3C standard, see http://www.w3.org/RDF/

[9] “Linked Data – Connect Distributed Data Across the Web” [Online]. Available: http://linkeddata.org

[10] Dieter Fensel et al., “Towards LarKC: A platform for web-scale reasoning,” *Proc. IEEE Int’l Conf. Semantic Computing*, pp. 524-529, 2008.

[11] E. Lehtihet and N. Agoulmine, “Towards integrating management interfaces”, *Network Operations and Management Symp*. pp. 807-810, 2008.

[12] M. Sethi et al., “An open framework for federating integrated management model of distributed it environment”, *Network Operations and Management Symp.*, pp. 803-806, 2008.

[13] D2RQ Platform, Freie Universität Berlin, Web-based Systems Group [online]. Available from: http://www4.wiwiss.fu-berlin.de/bizer/d2rq/

[14] Jena: A Semantic Web Framework for Java. [Online]. Available: http://jena.sourceforge.net/

[15] C. Bizer, T. Gauß, R. Cyganiak and O. Hartig, “Semantic Web Client Library” (2008). Available from: http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient.

IBM and Tivoli are trademarks of International Business Machines Corporation in the United States, other countries, or both. Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.