RZ 3752 (# 99762) 10/12/09
Computer Science 11 pages

Research Report

Fusio: Semantic Integration of Systems Management and Enterprise Information

Axel Tanner and Metin Feridun

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Email: {axs | fer}@zurich.ibm.com

Artem Nikulchenko*

National Technical University “Kharkiv Polytechnical Institute”
Frunze str. 21
Kharkov, 61002, Ukraine

Email: nikulchenko@gmail.com

*This work was performed while the author was at the IBM Zurich Research Laboratory

IBM Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

Fusio: Semantic Integration of Systems Management and Enterprise Information

Axel Tanner[1], Metin Feridun[1], Artem Nikulchenko[2*]

[1] IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland
{axs | fer}@zurich.ibm.com
[2] National Technical University "Kharkiv Polytechnical Institute", Frunze str. 21, Kharkov, 61002, Ukraine
nikulchenko@gmail.com

Abstract. The growing complexity of the typical enterprise IT environment requires better ways for managing the infrastructure. Often much information about the infrastructure is available, although in separate and incompatible systems management products. We present a prototype of an integration platform, called Fusio, that is based on semantic web and text-search technologies to bring this information into a common context. This approach is lightweight and flexible, and therefore enables the integration with other, nontraditional information sources.

1. Introduction

As IT systems grow in size, so does the complexity of managing them. Effective systems management relies on the availability of management data and operations from many and diverse sources. Explicitly declared or discovered relationships between data are essential in creating a good understanding of the management issues. In an example problem management scenario, a monitoring system issues an alert indication that a resource, e.g., a server has failed. The name of the server is matched against the application inventory database to determine which business applications are effected by the failure. This information helps assign a priority to the resolution of the problem. The next step is to find a specialist who can fix the problem, e.g., from a personnel database, and who is available, e.g., from a calendar application. Finally, the actual work is scheduled and tracked by issuing a trouble ticket. This scenario is just one example illustrating how diverse data originating from multiple systems can be utilized. Moreover, it highlights the need to increase the effectiveness of systems management and enable automation.

Although the benefits of data integration are well known, its implementation poses a number of challenges, leading to custom solutions in which the data is manually wired together. Although efforts were made to create standard domain models and interfaces such as SNMP [1], CIM [2], WS-Man [3], and WSDM [4], seamless integration remains elusive as there is neither a single, central model of management data

* Work performed while at the IBM Zurich Research Laboratory

that is universally accepted nor are standard interfaces available to access data held in management systems. Under such circumstances, creation of an integrated management solution requires knowledge of (a) the data models supported by the relevant data sources; (b) how data can be collected from each data source, and (c) the relationships between data collected from various sources. The level of expertise needed to build custom, integrated solutions and maintaining the solution as new data sources become available or the existing ones are modified are costly.

The goal of the research reported in this paper is a simple and cost-effective creation of integrated management solutions. Based on earlier work by the authors [5], in this paper we report on the data integration platform called Fusio, which uses a combination of semantic web [6] and traditional text-based search technologies to enable a loose coupling of semi-structured information based on source-dependent, differing data models. The ability to pull together information from any relevant data source, e.g., management products, calendars, databases, directories etc., and to extract and establish relationships between data within and across sources makes it possible to build management solutions with different perspectives and capabilities in an easy and quick fashion. For example, Fusio makes it possible to build applications that search across management data, and it simplifies the management of distributed applications such as business processes, which typically run on multiple systems. Integration is also relevant in the emerging cloud computing and "software as a service" paradigms, in which computing services are in effect a composite set of services, potentially coming from multiple providers.

The structure of this paper is as follows: in section 2, we describe the architecture of Fusio, followed in section 3 with a description of the prototype we have built to demonstrate key Fusio concepts. Section 4 is a summary of related work, and the paper concludes with a discussion and a description of our current and ongoing work on Fusio.

2. Fusio Infrastructure

Management data sources in actual IT environments in many cases do not share the same or compatible data model and the same methods to access data. Fusio therefore assumes a heterogeneous environment: in response to a query, it uses source-specific methods to collect data; converts collected data into an RDF [7] representation with little or no manipulation of the original data, and uses rule-based inferencing to refine and relate data from multiple sources. Text-based search and a Fusio data model are used to focus and simplify this process as described in more detail below.

Note that Fusio does not introduce an additional central repository of management information. Such a store would be potentially huge and would not be suitable to retain the transient data needed, e.g., CPU load right now. Therefore all information in Fusio is fetched on demand selectively driven by the user query, using partial caching for efficiency.

2.1 Technologies Used

The problem of data integration in systems management is similar to the problem of bringing together information in the World Wide Web in an automated way. To address this, Tim Berners-Lee et al. have created the vision of the *Semantic Web* [6], for which now there is support of established or emerging standards such as RDF/RDFS, OWL, SPARQL, etc. [8]. The Fusio project uses these standards as the underlying technology. It also benefits from the search technologies that have been developed to handle the vast amount of information in the Web: aside from highly structured semantic linking of data, Fusio also uses full-text indexing of the data to allow faster and easier search and additional interlinking between information elements.

2.2 Fusio Data Model (FDM)

The *Fusio Data Model* serves two purposes: one, it provides points of reference where the source-specific data providers can attach the information they are gathering. Two, it serves as a bridging or mapping model for applications to enable them to pose queries for which responses will span multiple data sources. Hence an application can ask about "performance" of a resource and receive all available information under that topic. A data provider can attach its performance-related information under the topic "performance", permitting information from different data models and providers to be brought into a common context even with a very high-level model. This kind of *loose coupling* will not aim at identifying and mapping every detail of the source models, but will allow quick and easy integration of new or changing information sources that can be refined when needed, e.g., by collective definition by the users.

A sample FDM is as shown in Figure 1. In this model, we represent a computing resource as a "ComputerSystem", and assign a number of sub-properties such as "Events" or "Network Topology". Each data provider registers as a "source" to any one of the properties for which it can provide data.

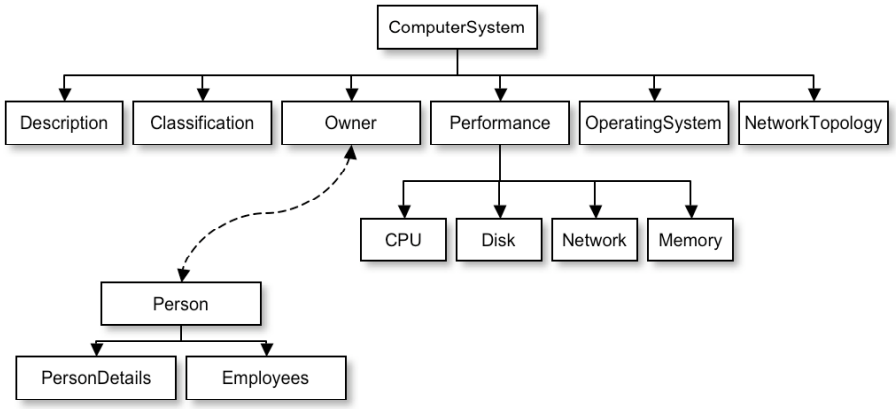

Figure 1 Sample Fusio Data Model

Figure 1 also shows how partial models can be linked: An "Owner" of a resource is linked to a "Person" as they represent the same concept, but have different names because they originate from different data domains ("owner of a resource" versus "a person, an employee"). Linking data in this way not only exploits explicit relationships, but can also develop implicit but useful relationships (e.g., an *owner*, who is a *person*, is *on vacation* and therefore *unavailable to fix the problem* now).

2.3 Architecture

As shown in Figure 2, there are three main components of the Fusio infrastructure: a *Query Manager* for handling queries received from applications; a *Search Index*, a set of indices that help initial finding of information and support query processing; and the *Fusio Engine*, which processes the queries by collecting data and creating the RDF response.

Search Index

The *Search Index* contains three separate indices that help focus the gathering of data by the Fusio infrastructure. In part, these are seeded up-front, but continuously added to and updated during query processing. The *Model Index* keeps track of which data provider provided information about an element in the model. The next two indices are built out of the RDF triples collected by the data providers. Given an RDF triple consisting of {subject, predicate object}, the *Identities Index* is built from the subject entries (i.e., URIs), and the *Values Index* is built from the literal values in the object entries. Given a query with the search term `"prugiasco* topics:cpu"`, it is possible to identify the data providers knowledgeable about the topic `cpu` from the first index, and in addition, find the instances matching our interest in `prugiasco*` from the identities or values indices, based on name or literal properties.

Query Manager

Given a query from a top-level application, the *Query Manager* calls the query analyzer to determine which data sources should be tapped to answer the query. The assembled set of queries is then sent to the *Query Processor*, which sends them to the *Fusio Engine* for processing. The result received as an RDF graph is then processed by the *Query Renderer* into the desired format (e.g., HTML or other formats for further handling in the calling applications) and sent back as a response.

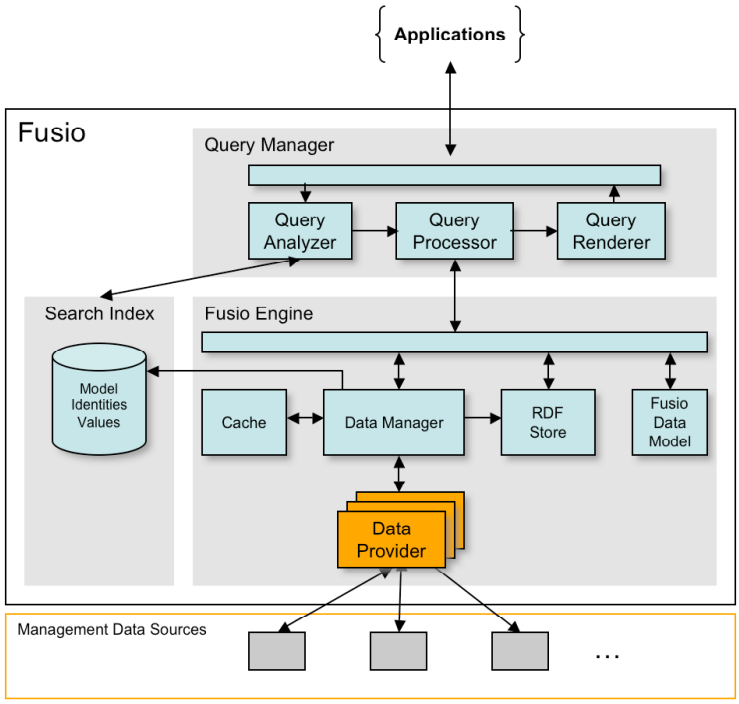

Figure 2 Fusio Infrastructure

Fusio Engine

The *Fusio Engine* acts as an intelligent broker that fetches on demand the information from only the relevant *Data Providers* for an incoming query. To achieve this, it uses the *Fusio Data Model* (FDM) to determine which Data Providers it has to query, based on which and how data providers have registered with the model. The *Data Manager* coordinates the queries to multiple data sources, and stores and combines the sets of RDF triples received from each Data Provider in a temporary *RDF Store*. To avoid repeated calls to data providers, retrieved data triples are stored in the *Cache*. As data will have different lifetimes, e.g., inventory information will usually be much slower to change than performance metrics, each Data Provider associates a lifetime for the data they provide to prevent "stale" data in the Cache. Based on this information, it is possible to decide whether the cached data can be used to satisfy a query or a refresh of the cached information is necessary (data with a "zero" lifetime will not be cached at all).

After running all separate queries to the Data Providers, the RDF Store will contain the results in the form of triples that are linked into the FDM. The Fusio Engine will hand this RDF Graph back to the Query Manager for rendering according to the needs of the top-level applications.

Data Provider

A *Data Provider*, as shown in more detail in Figure 3, provides the interface between the Fusio Engine and the individual (currently mostly non-RDF) Management Data Sources and is needed for every source of data.

In the first step, the *Data Collector* uses the source-specific query given by the Data Manager, which corresponds to the information element of interest as registered within the FDM, to fetch the information needed via the source-specific interface and data format. Depending on the data source, this could be, e.g., via SNMP, web services, SQL, LDAP, etc.

In the second step, the collected data is converted into a representation as RDF triples by the *RDF Generator*, with little or no change to the structure of the original data.

Lastly, in the third step, the *Inference Engine* uses data-source-specific *rules* to link the generated RDF triples into the bridging Fusio Data Model by adding more triples, e.g., stating that a resource of type "Owner" is also of the type "Person" in the FDM. The combined (or if necessary, subselection of) RDF triples are handed back to the Data Manager.

We expect that in the future data sources will be able to provide their data already in RDF data format, with its corresponding model described as OWL ontology. Therefore, in the Fusio prototype, the RDF Generators are quite low-level, i.e., they will in particular generate RDF triples using data-source-specific namespace and naming, thereby putting the task of integration into the FDM entirely into the responsibility of the Inference Engine driven by a set of corresponding, data-source-specific rules, i.e., step 3 above.

In the future when data sources will be able to provide data in RDF format, step 2 above is not needed, the Data Provider will only need to perform the linking and integration of the data-source-specific triples into the FDM, which in the language of the semantic web would correspond to a (partial) matching of different ontologies.

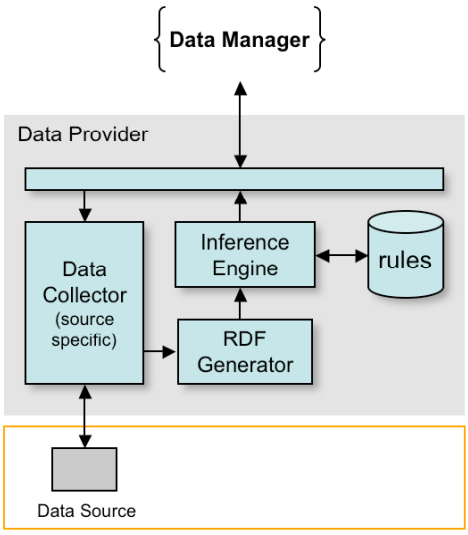

Figure 3 Details of a Data Provider

3. Fusio Prototype

To demonstrate the capabilities and the applicability of the Fusio architecture, we built a prototype that integrates information from an inventory system (a custom-built application); a general system management product (IBM® Tivoli® Monitoring, ITM); a network management product (IBM Tivoli Network Manager, ITNM), and an LDAP-based directory of employees. Other products have been integrated into the prototype, but are omitted here for brevity.

The Fusio prototype is implemented in Java™ and packaged as a set of servlets, utilizing the Lucene search engine [9] for the indexing and search functionality, and the Jena Semantic Web Framework library [10] for all RDF-related code, e.g., RDF creation, modeling, inferencing and querying. The fusio*explorer* front-end is implemented with current web technologies, such as AJAX and Dojo [11].

Figure 4 shows a screenshot of the servlet-based application "fusio*explorer*", which runs on top of the Fusio infrastructure described in the preceding section. In this screenshot, a system administrator needs additional information or details for a system approximately called *tarnolgia*, and uses the query '`tarnolgia*`' to invoke the fusio*explorer* to see what the underlying management systems know about this name.

The Fusio system searches the underlying index and finds entities that match this string in their name, identifier, or in one of their properties.

For each entity found, Fusio then fetches information from the data sources that are listed in the FDM as being able to get the data for this type of entity.

Information comes back via the Data Providers in the form of RDF triples all relating to instantiations of the FDM for the different entities. These entities are shown in

the fusio*explorer* GUI as different "tabs", with each tab displaying the returned information according to the structure of the corresponding part of the FDM.

In the specific example shown in Figure 4, we see that three different entities (corresponding to the three tabs) are found – one for the ComputerSystem tarnolgia.zurich.ibm.com; another tab because tarnolgia is known by the 'Itcs' inventory management system, and a third, because tarnolgia itself is the server running the ITM management system.

In Figure 4, the tab selected is for the ComputerSystem tarnolgia, showing the available information from different management systems in the hierarchy given by the FDM, e.g., description, ownership, classification, operating system from the inventory system 'Itcs', events, operating system and status from ITM, network topology information from ITNM and network performance information from ITM and Aurora [12]. The aspect of *loose coupling* can be seen here as two different sources (Itcs, ITM) that provide information about the operating system of tarnolgia, but use different literal strings to describe the same fact ("Windows® XP"). Currently, these are not reconciled but shown side-by-side.

Moreover, it can be seen that the *owner* information from the inventory system leads to the directory information (note: the local IBM directory of employees is called "BluePages"), showing *person* details for one of the authors.

In addition to what is shown in Figure 4, fusio*explorer* allows more complex queries such as '`prugiasco* and tarnolgia* topics: status and cpu`' to obtain only the status and cpu information for the machines prugiasco and tarnolgia because it understands these concepts from the FDM.

The "fusio*explorer*" application, a search engine for systems management information is of course only one demonstrator for the potential of Fusio. Other possible applications include:

- providing additional contextual information to complement data from existing management systems, e.g., add machine configuration history in trouble ticket systems, or machine performance data in network topology management systems;
- enriching system events on-the-fly with additional information, e.g., add to a "high-load" system event for a machine the information of the business processes this machine is involved in;
- giving system administrators and power users the ability to query for non-anticipated combinations of information for their own, possibly adding ad-hoc concerns into their own interfaces, e.g., in personal scripting, or feeding into RSS readers.

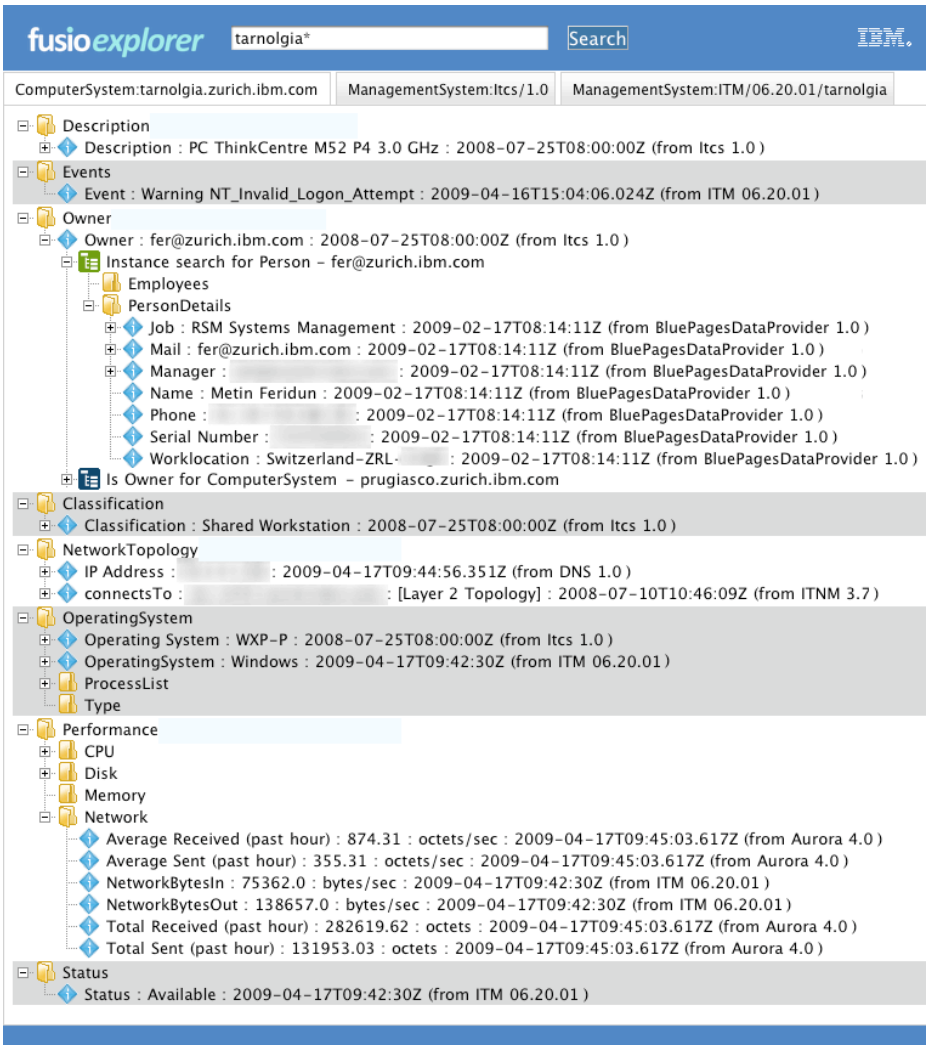

Figure 4 Example Application fusio*explorer*

4. Related Work

Integration of information from diverse sources is a long-running theme in systems management. The prevailing approach is to base integration on standard data models, such as SNMP, CIM and others, and to expect that all data sources conform to one or more of these models. This approach has had limited success over time as data sources change frequently and vendors compete and try to differentiate from each other, leading to nonstandard extensions and incompatibilities. In contrast, the semantic web approach used in Fusio fully embraces the "open world" assumption also un-

derlying the WWW, i.e., it is explicitly open to a flexible integration via model linking between differentiated domain specific models.

CMDBf (Federated Configuration Management DataBase) [13] is a recent effort to create a standard to link data from CMDBs from different vendors through reconciliation, but is largely limited to protocol and query language definition, rather than a true integration or federation of data into a common semantic level. Therefore it will not allow an easy building of queries across different data sources.

David Booth [14] describes and advertises the use of semantic web technologies sitting on top of systems management products, demonstrating it for HP's UCMDB product. We fully agree with this vision and show in this paper a practical prototype implementation of integration in a similar spirit with a broader set of management products, supported in addition by the introduction of full-text indices.

5. Summary and Discussion

We presented Fusio, a data-integration concept and prototype based on semantic web and text-search technologies. The key benefit of this approach is the loose coupling of information from diverse sources, which decreases the cost of creating integrated management solutions.This approach avoids the pitfalls of having to define one central model that encompasses all aspects of the underlying management systems, and instead restricts itself to the use of a simple top-level model for bridging purposes. This bridging model can be extended relatively easily to adjust to changing or new data sources.

Clearly, there is the trade-off of how many details to put into the bridging model: a smaller granularity allows the linking to (and querying of) finer details of the information given by the underlying data sources, but has more integration overhead. However, not all links between the information of different data sources have to be specified explicitly. The approach presented here facilitates also linking through non-structural means by using the literal information of the data sources in the indices.

Furthermore, the Fusio approach is lightweight as it fetches and federates information only when needed without a central store (although with caching).

A next step in the Fusio project is to look into extending the current platform to use the Linked Data [15] concepts. This will provide a more general combination of data in the semantic network and also enable the creation of a generic reasoning layer to establish connections between data. Another topic of current interest is in the design of tools to discover relationships between data from diverse sources to further ease the integration task.

References

1. Simple Network Management Protocol (SNMP), Internet RFC. See http://tools.ietf.org/html/rfc3411
2. Common Information Model (CIM), a DMTF standard. See http://www.dmtf.org/standards/cim/

3. Web Services for Management (WS-Man), a DMTF standard. See http://www.dmtf.org/standards/wsman/
4. Web Services Distributed Management (WSDM), an OASIS standard. See http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm
5. Metin Feridun, Michael Moser, Axel Tanner: “Building an Abstraction Layer for Management Systems Integration“, End-to-end Virtualization and Grid Management, Proc. 1st IEEE/IFIP Int'l. Workshop on End-to-end Virtualization and Grid Management "EVGM 2007," edited by M. Hasan, S. Figueira, Multicon Lecture Notes No. 7 (Multicon Verlag, Schöneiche, Germany, 2007)
6. Tim Berners-Lee; James Hendler and Ora Lassila: "The Semantic Web" in Scientific American, Vol. 284, No. 5, pages 34-43; May 2001
7. Resource Description Framework, a W3C standard to describe triples essentially stating subject-predicate-object statements. See http://www.w3.org/RDF/
8. W3C Semantic Web Activity, http://www.w3.org/2001/sw/
9. Apache Lucene Project, http://lucene.apache.org/
10. Jena Semantic Web Framework, http://jena.sourceforge.net/
11. The Dojo Toolkit, http://www.dojotoolkit.org/
12. Aurora Traffic Analysis and Visualization tool, http://www.zurich.ibm.com/aurora/
13. CMDB Federation Workgroup (CMDBf). See http://www.cmdbf.org
14. David Booth, “Enterprise Information Integration using Semantic Web Technologies”, presented at Semantic Technology Conference 2008. See http://www.semantic-conference.com/session/727/. Presentation available at http://dbooth.org/2008/stc/slides.pdf
15. Tim Berners-Lee: “Linked Data”, http://www.w3.org/DesignIssues/LinkedData.html