Searching Across the International Space Station Databases

David A. Maluf, Ph.D.
David.A.Maluf@nasa.gov
William J. McDermott
William.J.McDermott@nasa.gov
Christopher D. Knight
Christopher.D.Knight@nasa.gov
National Aeronautics and Space Administration
Code TI, Mail Stop 269-4
Moffett Field, CA 94035-1000

Ernest E. Smith
Ernest.E.Smith@nasa.gov
National Aeronautics and Space Administration
Code DA4, 2101 NASA Road 1
Houston, TX 77058

Mohana Gurram
mgurram@riacs.edu
Research Institute for Advanced Computer Science
Moffett Field, CA 94035-1000

Abstract— Data access in the enterprise requires us to combine data from different sources and different formats. It is advantageous to focus on the intersection of the knowledge across sources and domains; identifying appropriately-related knowledge and ignoring irrelevant knowledge which distracts and serves to make the integration unwieldy and more complicated than necessary. A *contextualized* search over multiple domains is proposed in this paper–useing context-sensitive queries to support disciplined manipulation of domain knowledge resources. The objective of a context search is to provide the capability to interrogate many largely semantically disjoint domain knowledge resources in an automated and intelligent fashion. The intelligent search system detailed supports formally the tasks of selecting, combining, extending, specializing, and modifying components from a diverse set of domains.

This paper demonstrates a new paradigm in *composition* of information for enterprise applications. In particular, it discusses an approach to achieve data integration across multiple sources in a manner that does not require heavy investment in application, database, and middleware maintenance. This *lean* approach to integration leads to cost-effective and scalable data integration with an underlying *schema-less* object-relational database management system. This highly scalable information on demand system framework, called NX-Search, is an implementation of an information system built on NETMARK. NETMARK is a flexible, high-throughput open database integration framework for managing, storing, and searching unstructured or semi-structured arbitrary XML and HTML used widely at the National Aeronautics Space Administration (NASA) and industry.

Keywords— Intelligent Information Systems, Netmark, NX-Search, Context Search, Semantic Interoperation, Heterogeneous Systems, Integration

TABLE OF CONTENTS

1. INTRODUCTION

A number of technologies have been developed to support large-scale interoperation among distributed information sources and applications such as databases and heterogeneous sources and applications on the Internet. However, managing large-scale interoperation of multiple *data sources* in an enterprise remains a task requiring many levels of expertise and an adherence to standards. Existing information systems and applications have strong notions of interfaces, often unique interfaces leading to increasing complexity of integration. Efforts like the HyperText Transfer Protocol (HTTP) and eXtensible Markup Language (XML, the next generation of HTML) are mechanisms to add semantics and structure and interoperability at fundamental levels to existing data repositories with minimal impact and simplified deployment and allow the specification of the domain's knowledge or the domain component syntax with the data.

Leading efforts in interoperating among multiple domains are often implemented as the *union* of multiple domains. An immediate, exponential increase in data integration is noticeable to data integration architects. Yet, there are negative economical consequences to the maintenance of the union of these interfaces. The union of multiple domains always turns out to be a low hanging fruit in the beginning...but always turns out not to be efficient for the evolving needs of customers and quickly becomes impossible to maintain. *System* evolution, the lifecycle from cradle to grave, now starts to become the most dominant question in the any investment of existing or new information systems.

This paper presents a solution to a problem of interoperability; the illustration for a benchmark is how to interoperate a Microsoft Word document with spreadsheet, PDF document, and a payroll relational database. Notably,

for interoperation and intelligent access to heterogeneous information, the focus should be on the *intersection* of the knowledge since intersection will define the required articulations. The term articulation refers to the linkages, which join concepts across domains.

The emergent need to define articulations between data resources has been in demand at NASA. We extend and generalize the identification of the articulation to a set of manipulations, such as selecting, combining, extending, specializing, and modifying components from diverse, common and domain-specific collection of sources. To deal with most semantic issues, a *context* and *content* domain selection is proposed. The intention is to support disciplined manipulation of resources. The representation of vocabularies and their structure is termed *domain knowledge* delineating the underlying contexts whereas the operations that combine and partition the domain knowledge in a sound and well-behaved manner are termed a *context algebra*. The basic algebra consists of three operations, namely intersection, union and difference (negation is considered an alternate form of the difference). Knowledge in this paper is limited to the knowledge that an expert can extract from a domain and not the domain itself (e.g. a complete schema dump.) The objective of a context-sensitive search is to provide the capability for interrogating many knowledge resources, which are largely semantically disjoint, but where articulations can be established on their perspective context. This paper describes the role of context search among multiple interfaces and demonstrates the use of an intuitive and flexible context algebra that provides users and system developers with the ability to intelligently manipulate components in real time.

2. BACKGROUND

The development of the mediation model reported in this paper is motivated by the need of interoperability among existing domain-specific representation of knowledge (structure and layout) and their respective formalisms (HTML, XML, Objects etc.). The spirit of this paper is to underline practical aspects of retrieving effectively data with an objective of matching or exceeding the agency current needs. However, what follows is a brief review of what is commonly known in the Knowledge Representation and information integration community [1][2].

The series of knowledge representation formalisms and frameworks starting with KL-One and currently culminating in systems and approaches like semantic-web provide powerful tools and knowledge expressiveness. However, they were intended to interoperate, but their complexity grows with the data. How much has to be added to their infrastructure and semantic-rich capability to achieve knowledge interoperability is still unclear. While knowledge representation is thought of as being a way to resolve integration problems, most knowledge representation formalisms have focused on paradigms, which assume an integrated environment and have been careless about managing the exceptions. In our approach, we focus on these exceptions.

From a research and technical point of view, there have been two recent efforts that open up possibilities for meaningful knowledge interoperation: the development of context logic and knowledge interfaces for sharing. The advance in context logic is the notion of translating encoded knowledge relative to its context and hence relates the knowledge to its domain. Advances in knowledge sharing revolve around translating knowledge from one formalism to multiple formalisms. However, the problem of translating many domains into different representations will create several problems. Semantic inconsistencies will arise from the terms and relationships used from the merged domains. Additional inconsistencies occur when the knowledge-content differs both in semantics and in compositional granularity and the union of multiple domain knowledge includes irrelevant knowledge and the result will be large, unorganized, and add disproportional cost to process.

With the success of Hypertext Markup Language (HTML) and large-scale content distribution of heterogeneous information, industry pushed the technology further with the eXtensible Markup Language (XML). XML is primarily intended to meet the requirements of large-scale Web content providers for industry-specific markup, vendor-neutral data exchange, one-on-one marketing, workflow management, the processing of Web documents by intelligent clients, and most meta-data applications.

The recent formal paradigm in the direction of porting knowledge from one representation language into multiple ones is the use of eXtensible Stylesheet Language Transformation (XSLT). For example XSLT is a mechanism for translating from one XML scheme and syntax into multiple-representation schemes. However, directly translating entire knowledge to any arbitrary representation leads to irrelevant knowledge and semantic inconsistencies, disproportioned in content. This mechanism requires tracking changes in the source scheme and output representations requiring a deep understanding of the XSL algebra and transformation paradigm.

3. APPROACH

Semantic interoperation became an industry fact with XML. XML is a system of standards and specifications that describe how software components, as being the domain knowledge, can interoperate across networks, languages and platforms. XML over the HyperText Transfer Protocol (HTTP) and related networking technologies allow for client-server interaction between heterogeneous objects distributed over a wide-area network; XML enables meta-information describing the semantics of objects in a system

and their interfaces available so that it can access other objects. Any object defined in XML can play simultaneous roles at the client and at the server. To reach effective interoperability with multiple databases with thousands of parameters, and for objects to plug and play, schemas have to be instantly discovered and integrated as part of the query and not hard-coded semantic mapping. The hard-coding paradigm of database integration schema is a significant burden and fatal flaw in today's integration approaches.

Linking the semantics across databases manually is a tedious engineering task with little scalability and high cost in maintenance. But maintenance is the intention of the integration providers. Reflecting on the complexity of the number of databases, engineering linkages is a prohibitive in cost and schedule. While it is in scope of a schema designer to craft a schema for a database, it is not justifiable to craft additional schemas to integrate heterogeneous databases; as the number of data sources is not bound to the designer. What works for ten may not extrapolate and work for hundreds or thousands.

The idea of combining *composition* and context discovery and binding with *declarative interfaces* is complementary. Declarative interfaces are primarily about specifying component context syntax and structure. C*omposition* binds the contexts into a new temporal subset of information through an intersection of the user query with data sources, followed by the union. The discovery mode is a promising outcome dealing with component design, component binding, and component semantics. Although simple in nature, this formalism is powerful enough to scale. A common MS Word document itself is treated as an independent source with same citizenship as full-scale relational database or XML document. Fundamentally, each contains source *contexts* that are based on their published interfaces.

4. ARCHITECTURE

The *context* and *content* mapping as well as the *context algebra* has been implemented at the National Aeronautics and Space Administration as an effort to solve an Information Technology (IT) challenge as well as a cultural challenge when dealing with the vast amounts of corporate data that NASA builds daily. The product tool suite is named NX-Search with two distinct components: *Composition* and the *Application* and a *Development Interface API*. The implemented system is named NX, a tool suite built on the NETMARK schema-less concepts. The original purpose of the NX-Search system and its Application Programming Interface (API) is to enable NASA information systems to do something that they could not do otherwise and to retrieve specific and precise information from within the contents of documents spread across disparate systems. This paper extends NX-Search to the community with the appropriate syntax and use cases. This demonstrates a custom NX-Search Query with a familiar user experience with minimal effort to use and maintain. The expectation, however, is to demonstrate NX-Search's capability to query unstructured information using standard set of programming patterns and practices. The use at NASA has been to retrieve data repositories based on both context and content, recompose new documents from the results of the queries, and publish information to the users in an intuitive, interactive, and iterative way. These demonstration queries combine relevant information from different sources into custom documents.

The immediate benefit from NX-Search has been to enable users and developers to select and integrate contents from proprietary electronic information software systems using standard interfaces and requiring little or no modification to the source repositories. NX-Search ability to design custom queries resolves the burden on existing systems to share information in a way not designed for originally. The primary audience for this technology has been primarily National Aeronautics and Space Administration (NASA). Nonetheless, industry and other NASA partners have adopted the system in their realm of products and services. This expansion has enabled a much larger community of developers or Help Desk engineers and IT engineers, creating, testing, and, troubleshoot queries that produce custom searches for work groups to meet a new generation of industry products and IT needs, with minimal effort. NETMARK, NX and NX-Search are standard-based application that enforces the World Wide Web Consortium Architecture Domain and Internet Engineering Task Force Standards, including the standards for Relational Databases, WebDAV, XML, XPath, [6][7][9].

The operation of NX-Search WebDAV HTTP API is shown in figure below

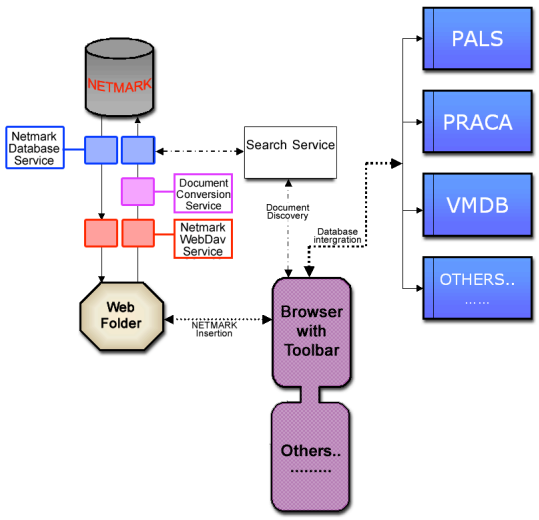

Figure 1 - Doc. Insertion, Discovery plus Integrating Multiple ISS Data Bases

The Context + Content Search take advantage of the boundaries that demarcate the location of information within a document. NX-Search Context Based Retrieval mechanism is illustrated in figure 2.

A section heading, such as "Procedure", that appears before a paragraph can throw light on the meaning of text within. Awareness of the meaning is conveyed by section-headings. Strong typed databases (semi-structured) use metadata information as the context for the query, thus enabling context plus content queries.

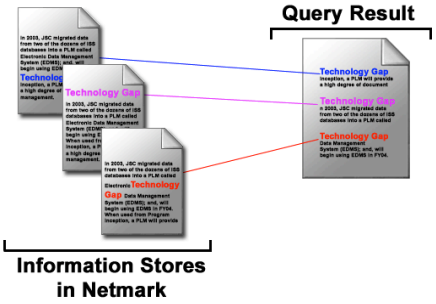

Figure 2 Context-Based Retrieval of relevant text fragments and documents

5. IMPLEMENTATION AND CASE STUDY

International Space Station (ISS) generates large amounts of change information about software that is distributed across multiple documents. This highly distributed information must be securely accessible and usable across the Enterprise. For example ISS Flight controllers needing this information face a challenge in finding the latest information on a pending change to ISS. A flight controller must query a range of document types stored in multiple databases using a different identifier each time. When something doesn't match, a flight controller has to sort through multiple of documents of different types to make the missing connection. Many tools has been developed to aid ISS flight controllers to perform integration and search across many types of documents across multiple sites and most of these tools have created bigger databases and/or large integrated schemas or semantic network. These schemes tended to become obsolete in short period of time as new resources emerge. To address flight controllers' requirements for fast and relevant search capabilities, the approach of dynamic database integration in a schema-less fashion have been developed. The Flight Controller leverages this new information storage and retrieval technology to accelerate time-intensive searches such as those performed by Station flight controllers. Furthermore, flight controllers require that search results be based on the latest documents stored in their systems of record, a requirement where most centralized and mediated systems fail due to a need of a common routing schema. As a matter of fact, our new system implements a periodical discovery and checking documents at authoritative sites for changes and updating cached documents with custom levels of currency and provides flight controllers with a level of assurance that information returned is based on the latest documents. This new approach tends to eliminate the human engineering bottleneck crafting the needed integrating scheme.

An aspect of this paper is to validate novel information integration capabilities in the context of high level of robustness flight environment and work-scenarios such as flight-controllers. These work scenarios dictate the document types and information sources that are constantly changing and updated. The solution to data currency provides assurance that search results are based on the latest documents from authoritative sources and provide also a long-term solution to data currency requirements. Additionally, replicating data from systems of record allow zero-cost integration of additional repositories and offload the impact of searching and retrieval from these systems. The focus of investigation, and of this paper, is on work scenarios that involve change documents stored in multiple databases. These databases have multiple document-types such Word, Excel spreadsheets, PDF documents, drawings, HTML and standard structured databases.

A typical Scenario

The Station Joint Software Review Board (JSRB) is responsible for managing ISS software changes to ISS. The JSRB uses a Software Change Request (SCR) form and a number of supporting documents to track software changes. The SCR and support documents contain well-defined inter-connecting links to one another. This rich network of inter-connected documents about a change is logically connected with a major ISS technical category. ISS core areas and other categories make up the structure of most information about Space Station. Station also uses this structure to organize ISS flight-controller groups. A flight controller is trained in one or more technical areas. Later, the flight controller is assigned to a group that is focused on a technical category in ISS structure. From the perspective of software modifications and work-around, flight controllers are primarily concerned with changes pertaining systems on ISS. Consequently, this paper focuses on scenarios that addressed change documentation about this technical area.

The scenario workflow of documenting a problem starts when ISS Engineering, NASA and its contractors document a problem in an Item for Investigate (IFI) and sends the IFI to the JSRB. The JSRB determines whether the item in question is a genuine problem that needs to be fixed. If so, the JSRB uses information in the IFI as the basis of a new Software Change Request (SCR). Although the SCR is a central document for tracking problem-resolution and deployment of software changes, the SCR and supporting

documents each contain links to one another, defining and connecting relationships between documents. Without some means of preserving relationships within and between documents, it is difficult and time consuming to glean all of the relevant information on an item in question. The number of external sources is not fixed and could spans many other database-like problem reporting databases (e.g. the Problem Reporting and Corrective Action System, aka PRACA), parts databases, drawing databases, procedure databases, etc.

Although the SCR is a central document for tracking problem-resolution and deployment of software changes, the SCR and supporting documents each contain links to one another, defining and connecting relationships between documents. Flight controllers have a particular interested in retrieving information that is distributed across a range of documents about a software change. For example, work-around, release-schedules and documentation changes about a software fix are documented in SCRs, Station Program Notes (SPN), and Schedule Issue Change Forms (SIF). A simplified diagram of the rich inter-relationships of documents in the system of record (PVCS) is shown in figure 3.

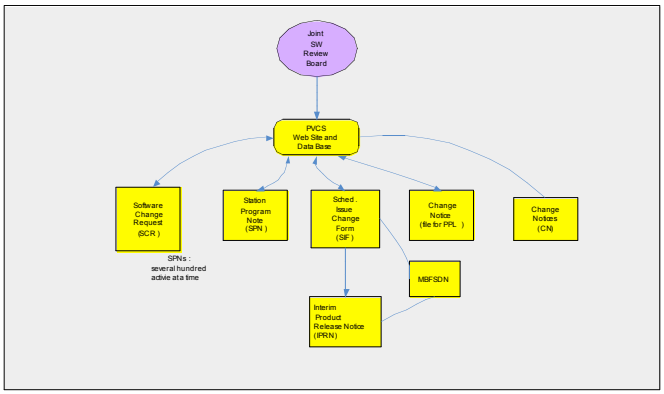

Figure 3 Inter-relationships between documents for SCR

Without some means of preserving relationships within and between documents, it is difficult and time consuming to glean all of the relevant information on an item in question.

Technology Gap

Software Change Requests and related change documents are accessible to flight controllers who seek to obtain the entire body of relevant information about changes but often face challenges. Change documentation is stored in a database that provides limited search capability. Flight controllers can use the existing search tool to query within a document-type: search for SCRs for example with keywords in TITLE attribute. A new query is needed for each source. It is very important to underline that the linkages among source is knowledge of the flight controller experience and not a simply a semantic map as it may seems. Search results of this title do not capture the relationship between the SCR that was found and its related, supporting documents. Flight controllers cannot search across document-types in a single query.

Querying multiple sources and documents

A scenario based on a search across multiple sources and documents is a requirement. The multiple source/document-types in this scenario on are listed in table 1:

Software Change Request (SCR)	Schedule Issue Change Form (SIF)
Station Program Notes, (SPN)	Change Notice (CN)

Table 1 Sample of Document-types in included in investigation

NX-Search is demonstrated as a multi-source integration tool, and has been turned over to flight controllers for validation. NX-Search user interface render as a web form with the capability query of "live data." Demonstration of queries of live data can demonstrate NX-Search fine-grain search-capability when querying change documents and specifications and also other notes, specifications, and reports from a range of information sources. Typically this underlines an integration of documents from multiple information sources matching different granularities.

The scope of information on software changes is not limited to change documents in multi-source environment. A sample of other information sources contain documents related to changes is shown in Figure 4.

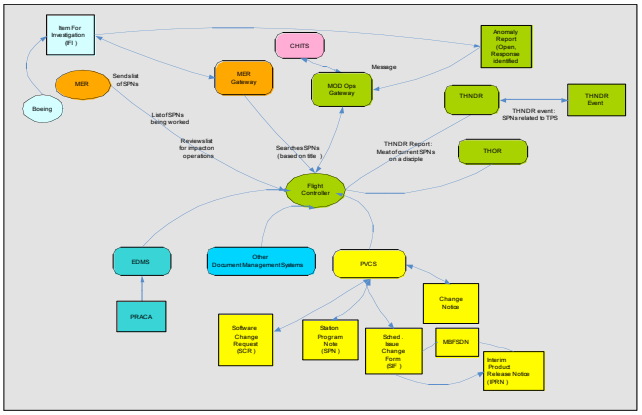

Figure 4 Sample of Document-types in PVCS included in investigation

The IFI database, CHITS, and DF's Anomaly Report contain information related to changes. To search across the range of document types stored in multiple systems requires software that is sensitive to relationships of sections within documents as well as relationships across documents.

Dealing with Asynchronous Updates

The high throughput of NX-Search back-end became also the solution to the data currency problem. This is performed by synchronizing updates to originals with documents being

cached. A periodical pedigreed cache becomes thus available. A logical representation of the update process is shown in figure 5.

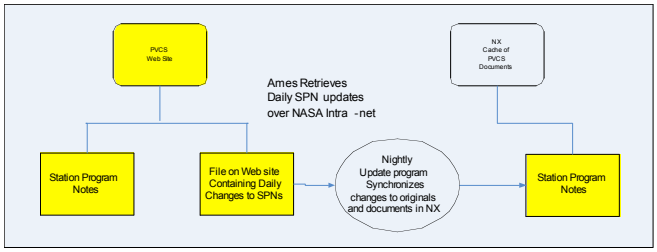

Figure 5 Mechanism for ensuring changes to SPNs in PVCS are reflected in documents.

The scalability of NX-Search has been applied to many document-types at ISS. Scalability NX-Search for ISS dataset is shown in Figure 6.

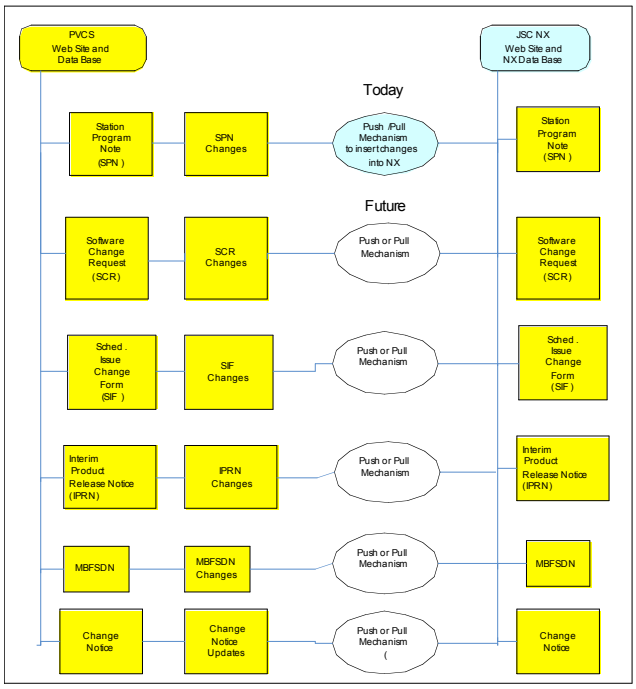

Figure 6 Generalizing the SPN update mechanism to other documents

The cache captures changes to SPNs and stores the information in NX-Search. The update reflects changes in these sources monitoring the evolution of the remote resources. However, the update routine is based on the same mechanism employed by ISS to update information on SPNs stored in the system of records to gain pedigree sign-off for the cached data. Systematically, search results based on the integrated view have the same fidelity to documents in their original sources. This degree of fidelity is highly important to the success of an information system when dealing with flight systems...producing a testable capability for synchronizing changes in multiple sources and NX-Search provides a firm line of evidence that the data currency problem is a tractable one.

REFERENCES

[1] A. Halevy et al., Enterprise Information Integration: Successes, Challenges and Controversies ACM SIGMOD International Conference on Management of Data, Baltimore MD 2005.

[2] D. Maluf, P. Tran, NETMARK: A Schema-Less Extension for Relational Databases for Managing Semi-structured Data Dynamically ISMIS 2003.

[3] J. Shanmugasundaram et al., SIGMOD Record 30, 20-26 (2001).

[4] H.V.Jagadish et al., VLDB Journal 11, 274-291 (2002).

[5] Oracle 9i Database Release 9.0.1 Developer Guide.

[6] J. Whitehead, Y. Goland, WebDAV: A network protocol for remote collaborative authoring on the Web CSCW 1999.

[7] XSLT, http://www.w3.org/TR/xslt.

[8] D. Maluf, P. Tran, T. La, "An Extensible 'Schema-less' Database Framework for Managing High-Throughput Semi-Structured Documents IASTED, Applied Informatics 2003.

[9] Xalan, http://xml.apache.org/xalan-j/.

[10] D. Draper, A. Y. Halevy, D. S. Weld:, The Nimble XML Data Integration System ICDE 2001.

[11] C. A. Knoblock et al., International Journal of Cooperative Information Systems (IJCIS) Special Issue on Intelligent Information Agents: Theory and Applications 10, 145-169 (2001).