Paper Number

An Extensible Information Grid for Risk Management

David G. Bell
Research Institute for Advanced Computer Science
NASA Ames Research Center, MS 269-4
Moffett Field, CA 94035-1000
USA

David A. Maluf
NASA Ames Research Center, MS 269-4
Moffett Field, CA 94035-1000
USA

ABSTRACT

This paper describes recent work on developing an extensible information grid for risk management at NASA – a RISK INFORMATION GRID. This grid is being developed by integrating information grid technology with risk management processes for a variety of risk related applications. To date, RISK GRID applications are being developed for three main NASA processes: risk management – a closed-loop iterative process for explicit risk management, program/project management – a proactive process that includes risk management, and mishap management – a feedback loop for learning from historical risks that 'escaped' other processes. This is enabled through an architecture involving an extensible database, structuring information with XML, 'schema-less' mapping of XML, and secure server-mediated communication using standard protocols.

INTRODUCTION

This paper represents the integration of two research activities within NASA: 1) risk management processes and tools, and 2) information grid architecture and implementation. The paper is organized to introduce these two research activities, and then the main section of the paper describes the research being completed to integrate them.

RISK MANAGEMENT

Since its inception in 1958, NASA has created a tremendous record of success which includes such publicly memorable moments as when NASA put a man on the moon in 1969 with Neil Armstrong saying "One small step for (a) man, one giant leap for mankind," when NASA launched the first reusable space launch vehicle in 1981, the Space Shuttle Columbia, and when NASA put a free-ranging robotic rover on the surface of Mars in 1997, the Rover Sojourner. NASA currently operates satellites in orbit around the earth to study phenomena such as El Niño, has had at least one astronaut living in the International Space Station since 2000, and continues to operate spacecraft to explore the universe such as the Voyager spacecrafts which are nearing the outer boundary of our solar system.

With all of its success, NASA has also had failures, which have cost billions of dollars and lost opportunity for scientific advancement, and more tragically have resulted in the loss of human life. Notable historical failures include the first manned Apollo flight in 1967 which resulted in three fatalities, the Space Shuttle Challenger launch in 1986 which resulted in seven fatalities, and the Mars Climate Orbiter and Polar Lander missions in 1999 which cost more than $1.5B. In the period of 1986 to 2001, the top ten NASA failures cost around $9.6B, half of that cost being due to the Space Shuttle Challenger [1]. NASA is not alone in experiencing such failures during this time period, with estimates of total U.S. space mission failures costing $18.6B and worldwide space mission failures costing $31.1B. Rates of failure for U.S. launch vehicles (NASA, DoD and Commercial) have been estimated to be 7.6% of the total number of missions for the period of 1985 to 1999 [2]. The costs of failure are high, and the rates of failure are not appreciably improving. Most notably in the last few years alone there have been a significant number of failures including Mars Climate Orbiter, Mars Polar Lander, and the Space Shuttle Columbia failure on reentry earlier this year.

Poor risk management and related human in the loop processes have been implicated as leading causes of these failures through both external and internal studies of NASA failures. Dianne Vaughan's historical ethnography of the space shuttle Challenger launch decision provides a detailed account that implicates a culture of normalizing engineering risk assessments over time, and implicates the organizational processes and structures related to risk assessment that in part shape the culture [3]. Internal analyses of the causes of failures as reported in NASA mishap reports have also implicated management processes and procedures related to risk management as a leading cause [4]. The lack of access to information and poor communication of information within those processes is specifically implicated. For example, with the Challenger mishap, the engineering team struggled to obtain access to existing anomaly information associated with the O-rings, and there was poor communication of related information between engineering and management, and between levels of management [3].

While NASA has extensive written procedures and guidelines for these human in the loop processes, recent advances in information technology research provide an opportunity to significantly improve them, to help the agency better manage risk for improved mission safety and success. One of these technologies is the concept of an Information Grid.

INFORMATION GRID

The Information Power Grid (IPG) is the National Aeronautics and Space Administration's (NASA) project for providing seamless access to distributed information resources regardless of location [5]. The overall project addresses three major categories of distributed resources: 1) hardware resources, such as super computers and scientific instruments, 2) software resources, such as simulation software and CAD programs, and 3) data resources, such as data archives and document databases.

While recent work in this area has focused on access to structured data archives, our focus is on integrating structured, semi-structured, and unstructured information; and doing this in a way that enables easier customization and integration of the technology for a multitude of heterogeneous work practices and related work processes [6]. As with many enterprises, information and information processes services at NASA are highly distributed. NASA and its contractors have hundreds of databases with millions of records and hundreds of desktop computers with millions of files. The formats and structures of the information are diverse with hundreds of file-types and hundreds of thousands of explicit and implicit structures. The decision making applications that utilize this information are numerous with hundreds of procedures and guidelines and hundreds of thousands of diverse work practices. An Information Grid can provide seamless integration of these distributed heterogeneous information resources for distributed heterogeneous scientific and engineering applications.

XDB-IPG is an open and extensible architecture being developed at NASA that enables the creation of such an Information Grid. XDB-IPG enables efficient and flexible integration of heterogeneous and distributed information resources using a novel "schema-less" database approach involving a document-centered object-relational XML database mapping. This enables structured, unstructured, and semi-structured information to be integrated without requiring document schemas or translation tables, which significantly improves its ability to be used to integrate across distributed heterogeneous scientific and engineering applications.

XDB-IPG utilizes existing international protocol standards of the World Wide Web Consortium (W3C) Architecture Domain and the Internet Engineering Task Force, primarily: 1) HTTP: Hypertext Transfer Protocol – a successful and stable request/response protocol standard, 2) XML: Extensible Markup Language – A ubiquitous five-year old standard that defines a syntax for exchange of logically structured information on the web, and 3) WebDAV – A widely supported four-year old standard that defines HTTP extensions for distributed management of web resources. While the third of these standards was primarily designed for distributed authoring and versioning of web content, XDB-IPG leverages WebDAV for management of arbitrary information resources including information processing services.

Through a combination of these international protocols, universal database record identifiers, and physical address data types, XDB-IPG enables desktop computers and distributed information resources to be linked seamlessly and efficiently into a highly scalable information grid. XDB-IPG is a flexible, high-throughput open architecture for managing, storing, and searching unstructured or semi-structured data. XDB-IPG provides automatic data management, storage, retrieval, and discovery [7] in transforming large quantities of highly complex and constantly changing heterogeneous data formats into a well-structured, common standard. Additionally, XDB-IPG is being integrated with more traditional collaborative information management technologies, to provide a range of functionality such as automatic notification, versioning for traceability, fine-grained access controls, workflow, and a range of other traditional functions that facilitate communication of information between machines, between humans, and between humans and machines.

MAIN SECTION

A RISK INFORMATION GRID is being developed by integrating this information grid technology with risk management processes for a variety of risk related applications. This section describes the related RISK INFORMATION GRID processes, applications, and architecture.

RISK INFORMATION GRID - PROCESSES

As a starting point for improving risk management using an information grid, this research is initially focusing on applying XDB-IPG for three related NASA procedures and guidelines (NPG) to create a RISK GRID: 1) NPG 8000.4 NASA Risk Management Procedures and Guidelines [8], 2) NPG 7120.5B Program/Project Management [9], and 3) NPG 8621.1 NASA Procedures and Guidelines for Mishap Reporting, Investigating, and Recordkeeping [10]. Together these three procedures and guidelines are meant to cover a significant part of NASA's policies for achieving program/project goals for mission safety and success.

NPG 8000.4 NASA Risk Management Procedures and Guidelines describes an overall closed-loop iterative process for risk management. Interestingly, the guidelines provide an inventory of sources of information relevant to a RISK GRID: people (e.g., team members, external experts, and external review boards), technical analyses of faults and failures (e.g., empirical test data and simulations), social and organizational analyses (e.g., of work breakdown structures and of resources/schedules), explicit risk data (e.g., risk mitigation/planned action milestone), and historical data (e.g., lessons learned and mishap investigation board reports). While NASA has had extensive processes for risk management, associated information technologies such as the NASA Problem Reporting and Corrective Action database were "clearly build in an earlier era before modern information technologies became available" [11]

NPG 7120.5B Program/Project Management describes a proactive approach for managing programs and projects for mission safety and success. Information addressed here includes the following: requirements management, independent review, work-breakdown structure, program and project management process metrics, and explicit risk management information. Information used for day-to-day program and project management provide a rich source of information for risk management, and the periodic independent reviews that are required as part of this process also provide one focal point for risk management that spans the lifecycle of programs and projects. Review practices are of particular interest here since they have been implicated in studies of NASA mishaps and anomalies. One major study reported that "inadequate review is a frequently cited theme in the recent mishap reports" [12], and another major study provided a case example where better review practices could potentially have mitigated risks [3]. Another study estimated that for a subset of post-launch problems/failures in the missions analyzed around 80% of the problems/failures "possibly could have been identified in the design review" [13].

NPG 8621.1 NASA Procedures and Guidelines for Mishap Reporting, Investigating, and Recordkeeping describes the feedback loop for cases where attempts to mitigate risks have failed resulting in a mishap. Starting as far back as the Apollo missions, whenever there has been a mishap or close call, there has been an investigation of the mishap. These investigations are typically conducted by a team of experts, and one output of these investigations is a document that includes the team's findings relative to the cause(s) of the mishap, and corrective actions that the team believes will help eliminate unnecessary risks associated with the cause(s) in order to improve future mission safety and success. While NASA has systematically investigated mishaps when they occur, documented investigation results in individual mishap reports, and subsequently enacted changes based on individual mishap investigations, NASA has not systematically analyzed sets of mishap reports, provided a ready means for people to make extensive use of mishap documents, or extensively improved mishap related methods and information technology systems.

A close look at the information used for these processes finds that the information is very heterogeneous and relatively unstructured, with much of the information residing in spreadsheets and desktop publishing reports that people mail to each other, rather than in structured databases. For example, a primary historical database of the costs of mission failures is a spreadsheet, explicit risk estimates for program products are authored in separate spreadsheets by the people responsible for each product, and desktop publishing forms are used to analyze fault trees in mishap investigations. The software systems used for these processes are varied with many locally customized applications written in various programming languages across a range of computing platforms. For example, tools for risk analysis used in conceptual design involve spreadsheet macros and Java programs. The social practices that make use of these tools are highly distributed raising communication challenges. For example, programs typically involve multiple organizations, reviews often span academia, industry and government, and program management spans multiple levels across NASA organizations.

RISK INFORMATION GRID - APPLICATIONS

The RISK GRID prototype is being used in a variety of applications for each of the three NASA procedures and guidelines described above.

Mishap Management – A feedback loop

Using mishap reports as a source of information about historical risks and their effects, the RISK GRID is being used to classify the causes of mishaps as reported in mishap reports, and then to complete trend analyses of causes over time. These trend analyses can be used by program managers to prioritize investments during the formulation phase of their program, and by independent reviewers during formal reviews to prioritize their questions related to technical as well as human and organizational risks. As currently implemented, the cause classification is done using a spreadsheet tool, and then published to an extensible database on the RISK GRID in XML format using standard HTTP protocols. Data mining tools are then used to conduct trend analyses on the classified causes, with options for plotting trends related to particular sets of subsystems or cross-cutting functions, and to particular sets of human and organizational causes. A mission failure cost database [1] has also been connected to the RISK GRID in a similar fashion. The mission failure database includes costs associated with mishaps, including both reference payload and launch costs, as well as direct and indirect costs of failure such as the cost of the mishap investigation and the "get well" program. Analysis tools are now being developed to integrate trend analyses across both of these datasets, and to extend the analysis capabilities for anomalies – mishap precursors.

Program Management – A proactive process

For program management, the RISK GRID is being used to allow individual product managers to document their self-assessments of risk in personal risk tracking spreadsheets, and then share them through the RISK GRID so that management can track and help mitigate risks across all products. Classes of risk covered include methods, data, software, integration and test environment risks. For each class of risk, the unmitigated risk effect is documented, along with mitigation strategy and implementation status. For both unmitigated and mitigated risks, self-assessments of likelihood that risk will occur and program level impacted are documented in order to track unmitigated and mitigated risk exposure, along with risk timing. To share the risk information, each spreadsheet has a short macro that utilizes the standard http interface protocol to publish an XML representation of the information stored in the spreadsheet into an extensible database on the RISK GRID. Management then uses their own customized interface to the extensible database to view all risk information for all products, sort by various categories such as mitigated risk exposure, and work to implement the risk mitigation strategy and reduce the likelihood of the risk effect. Additional functionality such as automatic notification and trending are planned to be added to this application as the tool use evolves.

Risk Management – An iterative closed-loop process

Explicit risk management of mission designs is also being explored for use with the RISK GRID. Here, an initial application is being developed to integrate historical mishap and anomaly information into risk tools used in conceptual design for managing risks and mitigations with project objectives [14]. This application is still in a formative stage, and one of the use scenarios is focused on supporting the use of the historical information for external reviews of designs. Additionally, the RISK GRID is being explored for sharing risk information within a project, across projects and programs, and across organizations associated with NASA enterprises. For this application, the fine-grained access controls and secure communication aspects of the RISK GRID are key capabilities for server-mediated peer-to-peer communication among risk tools and data resident on individual computers. Here, real-time use is a key requirement in an environment described as involving "extreme collaboration" [15] in design sessions, where local caches of data are used to eliminate time delays through communicating over the Web. Here the RISK GRID can support differential synchronization of risk information between local caches of information on individual computers and extensible databases of information on the RISK GRID.

RISK INFORMATION GRID – ARCHITECTURE

To enable these applications, an architecture is being implemented that enables easy integration of heterogeneous and distributed information resources with the RISK GRID. The architecture enables this through the development of an extensible database, by structuring information with XML, by using a "schema-less" mapping of XML, and by using the extensible database on servers to mediate secure communications between resources integrated as part of the grid using standard communication protocols of the World Wide Web.

Extensible Database

The architecture utilizes an extensible object-relational database model that represents a middle-ground between two opposing database technology research and development directions, namely the relational model originally formalized by Codd [16] in 1970 and the object-oriented, semantic database model [17][18]. The traditional relational model revolutionized the field by separating logical data representation from physical implementation. The semantic model leveraged from the object-oriented paradigm of programming languages, such as the availability of convenient data abstraction mechanisms, and the realization of the impedance mismatch [19] dilemma faced between the popular object-oriented programming languages and the underlining relational database management systems.

The object-relational database model (ORDMS) takes the best practices of both relational and object-oriented, semantic views to decouple the complexity of handling massively rich data representations and their complex interrelationships. ORDBMS employs a data model that attempts to incorporate object-oriented features into traditional relational database systems. All database information is still stored within relations (tables), but some of the tabular attributes may have richer data structures such as those stored in XML. As an intermediate hybrid cooperative model, the ORDBMS combines the flexibility, scalability, and security of using existing relational systems along with extensible object-oriented features, such as data abstraction, encapsulation, inheritance, and polymorphism.

Structuring information with XML

In order to take advantage of the object-relational (OR) model defined within an object-relational database system (ORDBMS) [20][21], a standard for common data representation and exchange is needed. Today, the emerging standard is the eXtensible Markup Language (XML) [22][23][24], commonly viewed to be the next generation of HTML for placing structure within documents. XML is both a semantic and structured markup language [22], and is a simplified subset of the Standard Generalized Markup Language (SGML) defined by the International Standard ISO 8879.

The basic principle behind XML is simple. A set of meaningful, user-defined tags surrounding the data elements describes a document's structure as well as its meaning without describing how the document should be formatted [25]. This enables XML to be a well-suitable meta-markup language for handling loosely structured or semi-structured data, because the standard does not place any restrictions on the tags or the nesting relationships. Semi-structured data here refers to data that may be irregular or incomplete, and its structure can be rapidly changing and unpredictable [25]. XML encoding, although more verbose than database tables or object definitions, provides the information in a more convenient and usable format from a data management perspective. In addition, the XML data can be transformed and rendered using simple eXtensible Stylesheet Language (XSL) specifications [23]. It can be validated against a set of grammar rules and logical definitions defined within the Document Type Definitions (DTDs) or XML Schema [26] much the same functionality as a traditional database schema.

"Schema-less" mapping of XML

Since XML is a document and not a data model, the ability to map XML-encoded information into a true data model is needed. This is enabled by employing a customizable data type definition structure defined by parsing dynamically the hierarchical model structure of XML data instead of any particular persistent schema representation. The customizable driver simulates the Document Object Model (DOM) Level 1 specifications [27] on parsing and decomposition of elements. The node data type format is based on a simplified variant of the Object Exchange Model (OEM) [28] researched at Stanford University, which is very similar to XML tags. The node data type contains an object identifier (node identifier) and the corresponding data type.

The markup language parser is designed to be independent of any particular XML document schemas and is termed to be schema-less. The data within the XML documents is a tree of objects that are specific to the data in the document [26]. The parser models the document itself (similar to the DOM), using the same object tree structure for all XML documents. This is very different than the traditional object-relational mapping from XML to relational database schema models. In the traditional model, element type with attributes, content, or complex element types are generally modeled as object classes, the classes are mapped to tables, scalar types are mapped to columns, and object-valued properties are mapped to key pairs (both primary and foreign). The mapping used in this architecture eliminates the complexity inherent in the traditional model which requires different object tree structures for each set of XML documents.

Secure server-mediated communication

WebDAV is an extension of the http request-response protocol that utilizes XML as the syntax for communication, and that enables distributed management of web resources. WebDAV is broadly supported in modern programming languages and operating systems, and provides a common protocol for human-computer communication as well as computer-computer communication. Combined with Secure HTTP, these protocols enable secure server-mediated communication for the risk information grid. Server-mediated communication is important here since communication across various firewalls is a needed, and since a pure peer-to-peer architecture would require each computer on the grid to be accessible through the firewall which raises security issues.

The extensible database servers are used to mediate secure communication within the RISK GRID, providing a number of common grid services such as the following:

- copying, moving and organizing resources through hierarchy and network relations
- seamless access to information in diverse formats and structures
- automatic decomposition of information into a query-able XML database
- storing and retrieving information about resources using properties
- locking and unlocking resources to provide serialized access
- getting and putting information in heterogeneous formats
- context+content querying of information in the XML database
- sequencing workflows of information processing tasks
- a common protocol for human and computer interface to grid services

CONCLUSION

The RISK GRID utilizes an extensible database architecture with standard communication protocols to improve communication of risk information across NASA. To date, RISK GRID applications are being developed for three main NASA processes: Risk management, program/project management, and mishap management. The first of these is an iterative closed-loop process for explicit risk management, the second is for proactive management of programs and projects including risk management, and the third of these provides a feedback loop for learning from historical risks that resulted in mission failures. Together, the range of these applications that integrate heterogeneous risk information distributed across NASA is beginning to create a baseline for risk mitigation across the agency.

ACKNOWLEDGMENTS

XDB-IPG is the derivative of the NASA Information Technology-Base program, now part of the overall NASA Computing, Information, and Communication Technologies (CICT) program. The application of XDB-IPG to enable an Information Grid for Risk Management is funded by the NASA Engineering for Complex Systems program.

REFERENCES

1. Failure Cost Database prepared for the NASA Engineering for Complex Systems program by Science Applications International Corporation (SAIC), 2002.
2. M.T. Gaunce, M.T., D.M. Bergner, D.M., and A. Wong, "Design for Safety: NASA is Serious About Risk Assessment", Proceedings of the ESREL 2001 Conference, Torino, Italy, September 16-20, 2001, Volume 2, page 1195, ISBN 88-8202-099-2.
3. D. Vaughan, The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA, 1996.
4. T. Panontin, Mishap Cause Classification Report, 2003 prepared for the NASA Engineering for Complex Systems Program.
5. NASA Information Power Grid (IPG);http://www.ipg.nasa.gov/.
6. D.M. Maluf, D.G. Bell, C. Knight, P. Tran, T. La, J. Lin, B. McDermott, B. Pell, “XDB-IPG: An Extensible Database Architecture for an Information Grid of Heterogeneous and Distributed Information Resources.” Presented at the Workshop on Grid Applications and Programming Tools during the Eighth Global Grid Forum, June 25, 2003.
7. D. A. Maluf and P. B. Tran, “Articulation Management for Intelligent Integration of Information,” IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, Vol. 31, No. 4, pp. 485-496, November 2001.
8. “NASA Procedures and Guidelines for Mishap Reporting, Investigating, and Recordkeeping,” NPG 8621.1, June 2, 2000.
9. “NASA Risk Management Procedures and Guidelines,” NPG 8000.4, April 25, 2002.
10. “Program/Project Management,” NASA Procedures & Guidelines (NPG) 7120.5B, November, 21 2002.
11. H. MacDonald, in testimony to the Columbia Accident Investigation Board, March 6, 2003.
12. NASA Chief Engineer and NASA Integrated Action Team, “Enhancing Mission Success – A Framework for the Future,” December 21, 2000.
13. J.D. Quinn, “Flight P/FRs and the Design Review Process,” JPL Reliability Interim Significant Report D-11381, January 1994.
14. S.L. Cornford, M.S. Feather and K.A. Hicks: “DDP – A tool for life-cycle risk management”, *Proceedings, IEEE Aerospace Conference*, Big Sky, Montana, Mar 2001, pp. 441-451.
15. G. Mark, “Extreme Collaboration”, *Communications of the ACM. Vol. 45(6), pp.89-93, 2002.*
16. E. F. Codd, “A Relational Model of Data for Large Shared Data Banks”; Communications of the ACM, Vol. 13, No. 6, pp. 377-387, June 1970.
17. R. Hull and R. King, “Semantic Database Modeling: Survey, Applications, and Research Issues”; ACM Computing Surveys, Vol. 19, No. 3, pp. 201-260, September 1987.
18. A. F. Cardenas and D. McLeod (Editors), “Research Foundations in Object-Oriented and Semantic Database Systems”; pp. 32-35, Prentice-Hall, 1990.
19. J. Chen and Q. Huang, “Eliminating the Impedance Mismatch Between Relational Systems and Object-Oriented Programming Languages”; Monash University, Australia, 1995.
20. R. S. Devarakonda, “Object-Relational Database Systems – The Road Ahead”; ACM Crossroads Student Magazine, February 2001.
21. M. Stonebraker, “Object-Relational DBMS - The Next Wave”, Informix Software (now part of the IBM Corp. family), Menlo Park, CA.
22. E. R. Harold, “XML: Extensible Markup Language”; pp. 23-55, IDG Books Worldwide, 1998.
23. Extensible Markup Language (XML) World Wide Web Consortium (W3C) Recommendation, October 2000, http://www.w3c.org/TR/REC-xml.
24. “The XML Industry Portal”; XML Research Topics, 2001, http://www.xml.org/xml/resources_cover.shtml.
25. J. Widom, “Data Management for XML Research Directions”; Stanford University, June 1999, http://www-db.stanford.edu/lore/pubs/index.html.
26. R. Bourret, “Mapping DTD to Databases”; O’Reilly & Associates, 2000, http://www.xml.com/pub/a/2001/05/09/dtdtodbs.html.
27. L. Wood et al., “Document Object Model (DOM) Level 1 Specification”, W3C Recommendation, October 1998, http://www.w3c.org/DOM/.
28. R. Goldman, S. Chawathe, A. Crespo, and J. McHugh, “A Standard Textual Interchange Format for the Object Exchange Model (OEM)”; Database Group, Stanford University, 1996, http://www-db.stanford.edu/~mchughj/oemsyntax/oemsyntax.html.

CONTACT

The first author is a senior scientist working in the Collaborative & Assistant Systems area at the NASA

Ames Research Center, and is employed in the Human Centered Computing area of USRA/RIACS. Prior to working at NASA, he worked for ten years in the Systems & Practices Lab of the Xerox Palo Alto Research Center, and held an appointment as visiting lecturer at MIT where he managed one of the four research programs in the MIT Center for Innovation in Product Development. The first author can be reached at dbell@arc.nasa.gov.