Integrated Data Management within the Organization

Managing and Organizing Data for Assurance

Andrew Bruce, CISSP, PMP, FITSP-D
CTO, RiVidium Corporation (http://www.rividium.com/)
andy.bruce@rividium.com
21 July 2010

Topic Summary:

- The problem of managing data in a networked environment
- Data auditing and consolidation challenges
- Handling data disclosure concerns and planning for the future

Section 1: Introduction

Today, organizations all share a common challenge in managing and integrating data into a coherent whole. Data does little good unless it is available to support key decision makers as they guide the forward path of the enterprise. In this paper we look at some data integration challenges especially as they pertain to an organization's need to achieve the full line-of-sight into all required data stores necessary for the best business decisions to be made.[1] The classic CIA Triad requires confidentiality, integrity, and availability for data and information stored by an organization,[2] and we consider how the following data management issues can impact the Triad:

- **Information Islands:** Useful business data must be *available* to decision makers. We examine our own isolated business data and discuss how to tie that data into our overall information management framework.
- **Investigations in a Vacuum:** When different groups make copies of and perform analysis on centralized data, *integrity* problems can occur. We look at where this occurs within our own organization and our response.
- **Audit Dilemmas:** Another *integrity* problem exists when "authoritative" data actually exists as an information island; different groups using this data for analysis can result in contradictory answers to the same question. We examine how auditing tools can be used to uncover these islands.
- **Disclosure Protection:** Today, it is trivial to copy vast amounts of data surreptitiously using any number of techniques. We examine specific methods for protection against this attack on data *confidentiality.*

We close this paper by looking at the effect of emerging trends upon data integration.

Section 2: Information Islands

Section 2.1: Overview

The dictionary defines "information island" as "a body of information (i.e. electronic files) that needs to be shared but has no network connection."[3] We suggest an alternative: "a body of information unavailable for integration into the business decision-making process." Our definition acknowledges that for data to be useful, it must be *available* in the *right context,* at the *right time,* for the *right people*. Otherwise, an information island results.

Section 2.2: The Spreadsheet

Consider the lowly spreadsheet: its very ubiquity conspires with its low learning curve and high customization capability to make it a pernicious adversary against an organization's data integration efforts. We see significant amounts of corporate data and knowledge that exist only as an Excel sheet on someone's workstation. This can have legal implications for an organization's regulatory requirements such as Sarbanes-Oxley.[4]

In our own organization, we have a dedicated Business Development group that frequently creates Requests for Information (RFIs) for prospective customers. Each of these RFIs requires its own corporate capability set — all of which are invariably created within individual spreadsheets. We mitigate this problem by requiring the use of our corporate collaboration portal for storing these spreadsheets, but the problem of *effective* integrated search (beyond the semi-structured data in the spreadsheets themselves) continues to be an issue. As we develop more RFIs we find it increasingly harder to present a consistent overall corporate message, simply because we have so many spreadsheets to consider, review, and manually correlate.[5]

Section 2.3: Local Developer Boxes

The stereotypical software developer is a lone individual, surrounded by pizza and soda, working feverishly on "her" machine with "her" source code (visitors most definitely not welcome). In our development group, we write a tremendous amount of software for multiple simultaneous development efforts targeting different operating systems. This can easily result in each project being dedicated to individual developers and workstations (leading to information islands for the software making up the systems under development). These islands result in a lack of communication between different groups, duplication of efforts as the same types of problems are solved in isolation, and real danger to the corporation in terms of loss of intellectual property. In this case, data integration really becomes business-critical: it affects the organization's bottom line.

We address these problems proactively, let's cover just a few strategies here:

1. **Standardized Environments**. For all new projects, we build a baseline development machine image. Developers work from that defined image, which ensures that we understand exactly what that project requires. This approach obviates any perceived need for standalone development and helps promote integrated development efforts.

2. **Shared Storage.** Most of our development efforts require our developers to store their source data files on networked shared drives; these drives are automatically included in the organizational backup policy. This provides a good baseline to recover from catastrophic failures.

3. **Source Control**. We define an *appropriate* source control system for each project (not every project can use the same system). All development for a given project occurs within the context of that source control system, allowing us to generate code quality metrics automatically, to categorize coding problems as they are detected, and to help promote an overall integrated development environment.

Section 3: Investigations in a Vacuum

Consider: two different groups are tasked to address a similar problem as part of a larger effort. For example, within the Army both the Blue Force Tracker and Army Knowledge Online programs provide a *Chat* service. Both an enterprise portfolio analysis team and a technology assessment team may need to

identify such duplicate activities as a project deliverable. The problem here becomes *authoritative data*. In the case of the Army, there exists a master list of registered programs[6] that all system owners and custodians are supposed to update. However, the vast majority of programs update this list sporadically at best; in effect, the list functions like a global corporate spreadsheet. In this scenario, one must consider not only the trustworthiness of the people making the original entries but also that data consumers have no way of being notified as changes are made or new systems are added.

In our notional case we have two groups pulling "authoritative" data from the master list. These groups proceed to perform their individual analyses, disjoint not only from each other but from any knowledge that the master list has been updated. This can easily result in contradictory findings between the two groups (such as which programs of record provide a given technology capability). In a Kantian way,[7] this problem can extend itself through time as these mutually exclusive results themselves become authoritative sources for other reports, thus embedding wrong assumptions in more and more derived reports. Just as a small initial error can result in tremendous divergences when firing rockets,[8] our conflicting reports work over time to provide ever more skewed pictures of the organizational data they purport to depict.

Mitigating this challenge to data integrity requires the organization to invest heavily in identifying shared data sources and defining *ahead of time* how updates to these data sources populate back to data consumers. The Web service publication / subscription (pub/sub)[9] model offers the best way to prevent this type of problem from occurring.

Section 4: Audit Dilemmas

Section 4.1: Overview

In this section, we ponder how we can assist decision makers in answering sometimes poorly formed questions based on a lack of access to all the data (information islands) that may be needed to understand a problem. Seeing that technology is somewhat to blame by allowing individuals and groups to create standalone data sources, it seems only fitting to look to technology to provide the "remedial remediation"[10] necessary to build a solution.

Section 4.2: Crawlers

Consider a common problem — information can be stored on multiple independent systems that may be of use to decision makers. Some of these systems may be internal to the organization, and some may be external; as an example, consider marketing data such as the buying habits of customers over the last six months. The actual purchase information is available in local corporate data stores, while information such as overall census data is available from publicly available data sources. "Crawlers" exist to address this problem.[11]

A crawler is a type of program that can search a data source for information that may be of interest to the searching party. These crawlers come in all forms; some (Web crawlers) are used to read Web pages from millions of organizations' Web sites based on keyword scanning. (Google, Yahoo, and Microsoft all

use advanced Web crawlers to enable the Internet-based searching we depend upon every day.) Other crawlers can be configured to search vastly different types of data sources ranging from corporate databases to isolated disk documents (e.g. the spreadsheets we discussed earlier). The SharePoint application server provides a reasonably sophisticated crawling capability that can be programmed to scan multiple data sources and analyze semi-structured data such as Word documents and Excel spreadsheets.

As part of their goal to find and correlate data for support of *ad hoc* searches to be performed at a later time, all crawlers have certain features in common:[12]

1. *Central repository* – This stores search references, which include the original data locations, keywords extracted from the data, and semantic information such as the type of system the data was found on (e.g. a personal system vs. a corporate data server).

2. *Correlation engine* – Once data has been gathered, it must be indexed and correlated to some degree (otherwise, given a sufficiently large set of crawled data, performing searches would not be possible in a timely fashion). During this indexing and sorting, data origins (provenance) must be calculated to determine whether the crawler sufficiently trusts a given data source to include it in search query responses.

3. *Semantics* – Even where not explicitly identified as such, every crawler includes some variant on semantic reasoning. For example, the original Google search engine used a concept of "forward references" to associate particular matched words with Web pages, and these Web pages were themselves matched with other keywords based on previous searches (via concepts such as *page ranking*). The net effect is that searches and results do not exist in a vacuum, but affect each other in an almost neural-network kind of relationship. Tseng notes that this provides the semantics used to make searches more context sensitive for the user performing the search.[13]

While crawlers do not solve the underlying problem of *lack of communication* that information islands represent, they are an excellent tool for discovering and integrating isolated data sources.

Section 5: Disclosure Protection

Section 5.1: Overview

In keeping with our emphasis on data integration and the avoidance of information islands, we must discuss a natural outcome of improved data access; namely, that of disclosure (the opposite of confidentiality).[14] As data becomes more available, it behooves the security officer to ensure that controls are in place and capable of protecting the (newly available!) information.

Section 5.2: Tools to Prevent Disclosure

In this section, we identify three common tools used to prevent disclosure and our use of each in our own organization.

Section 5.2.1: Strong ACLs

Access-control lists (ACLs) exist as part of an access-control matrix (ACM) that defines an *object* (entity to be acted on, such as a computer or file), a *subject* (entity desiring an action), and the set of rights that the subject has over the object.[15] While ACMs define the simplest way to assign rights, for our purposes we consider ACLs to include not only ACMs but the entire set of access-control models (discretionary, non-discretionary or role-based, and mandatory).[16] Our point here is that a data owner must examine, categorize, and classify the data under her jurisdiction and ensure that sufficient logical controls are applied by the data custodian.

Section 5.2.2: Thin Clients

In our own organization, one effective method we use for preventing data disclosure is simply by eliminating the PC itself! Instead of the expected laptop or desktop computer at individual workstations, we have small devices that connect directly to our virtual machine (VM) manager. Users insert their smartcard (via our public key infrastructure, or PKI) and logon not to the local thin client but to the remote virtual machine. By taking this simple approach, we not only save system management costs (fewer physical PCs) but we also preclude the use of portable high-capacity storage devices (like writable compact disks and memory sticks). In conjunction with an official security policy signed by each employee and clearly identifiable logon banners, our employees all realize that data on company machines is indeed company property.

Section 5.2.3 Application-level Gateways (firewalls)

In our organization, all access to the Internet is guarded by a corporate firewall that runs at the application layer (Layer 7 in the OSI Model).[17] In practice, this means that our corporate Internet gateway analyzes all details of network traffic ("peeling back the layers" of Internet messages to analyze contents and attachments of email as well as Web requests). While there are serious privacy concerns with this approach, we ensure that our company security policy clearly informs our users that all network traffic (including email) can and will be scanned. We use this capability to check for (and deny) unauthorized data transfers to collaboration Web servers such as Google Sites.

Section 6: Data Integration into the Future

In this paper we've looked at the problems relating to information islands and how those problems affect an organization's ability to integrate its data. Data integration is a key component of enabling corporate decision makers to have access to and knowledge of *all* the data elements making up the corporate data landscape, resulting in more informed decision making.

However, this view of data integration still lies squarely in 20th century thinking. Efforts like Internet 0 (Internet Zero) describe a world where all devices are internetworked and all data is available in easily locatable and searchable *civic commons*.[18] "Entities" (devices and software) ultimately become enabling agents for end-users, operating on the user's behalf and communicating dynamically with each other. In this world, the big problem will be managing the *elimination of the data boundary*. To support extreme degrees of interoperability, all organizations (government, military, and commercial) will feel pressure to

allow significantly expanded levels of data access. Those organizations unable to respond to ever-changing and unanticipated data queries from customers (and their agents) will find that they are treated like closed networks are treated today (i. e. censorship): queries will simply be routed around them. The online world will dismiss these organizations as being irrelevant in the increased pace of business decision making.

Our theory is that over a fifteen year window, the current generation of socially-networked employees will define successful organizations as those that have identified and eliminated their information islands by carefully categorizing and classifying their data to allow maximum visibility while still minimizing risk. Only thus can the organization of tomorrow provide the minimum level of data access necessary to remain competitive while continuing to meet required privacy, regulatory, and proprietary requirements. Our challenge as Information Assurance professionals will be in finding the balance necessary between openness and protection.

Reference List and Endnotes

Birman, K., T. Joseph. "Exploiting virtual synchrony in distributed systems" in *ACM SIGOPs Operating Systems Review*. Association for Computing Machinery, 1987.

Bosworth, Seymour, M.E. Kabay, Eric Whyne, eds., *Computer Security Handbook: Volume 1, 4th ed.* Hoboken, NJ: John Wiley & Sons, Inc., 2009.

Coon, Clifford. "Ensuring Spreadsheet Accuracy in the Sarbanes-Oxley Era" in *IT Audit*. The Institute of Internal Auditors, 2005.

Harris, Shon. *CISSP All-in-One (AIO), 4th ed.* New York: McGraw-Hill, 2007.

Garfinkel, Simson. "History's Worst Software Bugs" in *Wired.com*. San Francisco, CA: Drew Schutte (publisher), 2005.

Howe, Denis. *Dictionary.com*. http://dictionary.reference.com/browse/information island (accessed: June 29, 2010).

Kant, Immanuel (trans. Norman Kemp Smith). *Critique of Pure Reason* (New York: Random House, 1958).

Krikorian, R., N. Gershenfeld. "Internet 0 – inter-device internetworking" in *BT Technology*. London, England: British Telecommunications, 2004.

Levinson, Paul. "Chapter 10 – Remedial Media." *The Soft Edge: A Natural History and Future of the Information Revolution* (New York: Routledge, 1997).

Pfleeger, Charles P., Shari Lawrence Pfleeger. "Chapter 7: Security in Networks." *Security in Computing, 3rd ed.* (Upper Saddle River, NJ: Prentice Hall, 2003).

Sergey, Brin, Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine" (Stanford University InfoLab, Computer Science Department, Stanford University). Retrieved on June 30, 2010 from http://infolab.stanford.edu/~backrub/google.html.

Stephenson, Peter R., ed. *Information Security Essentials: Section 1*. Auerbach Publishing, ISBN 978-1-4398-0030-0, 2009. Retrieved on June 6, 2010 from https://norwich.angellearning.com/AngelUploads/Content/MSIA_2_0/_assoc/msia_s1/msia_s1_readings/ise_section_1_et.pdf.

Tseng, H. Chris. "Internet Applications with Fuzzy Logic and Neural Networks: A Survey" in *Journal of Engineering, Computing, and Architecture* 1, no. 2 (2007). This article is available online from the publisher at http://www.scientificjournals.org/journals2007/articles/1237.pdf (accessed: July 1, 2010).

Zhu, Zhengxiang, Wuqi Song, and Jifa Gu. "A Multi-agent and Data Mining Model for TCM Cases Knowledge Discovery" in *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*. IEEE Computer Society, 2008.

---

[1] Peter R. Stephenson, ed. *Information Security Essentials: Section 1* (Auerbach Publishing, ISBN 978-1-4398-0030-0, 2009), pg. 175. Retrieved on June 6, 2010 from https://norwich.angellearning.com/AngelUploads/Content/MSIA_2_0/_assoc/msia_s1/msia_s1_readings/ise_section_1_et.pdf. We took our wording for the "line of sight" directly from this chapter. The chapter itself ("The IA^2 Framework") discusses how an organization's overall framework (in this case, related all the way to the Federal Enterprise Architecture or FEA) enables organization decision makers to see clearly how their overall systems tie together. Since information islands effectively hide knowledge from these decision makers, these islands must be found and eradicated. Thus, eliminating information islands is an integral part of any enterprise architectural effort.

[2] Seymour Bosworth, M.E. Kabay, Eric Whyne, eds., "Chapter 3.1: Proposal for a new Information Security Framework," *Computer Security Handbook: Volume 1*, 4th ed. (Hoboken, NJ: John Wiley & Sons, Inc., 2009), pg. 97. While in this paper we discuss the CIA Triad, the chapter cited is actually Mr. Donn Parker's *Hexad* that adds three "missing" elements to the Triad (this addition is the subject of some controversy in online discussions). Of note is that Mr. Keith D. Willett (author of *Information Assurance Architecture*, Chapter 2 of which makes up a reading from our *Information Security Essentials* textbook cited above) actually adds another three elements to the mix — making a "neuftet" in all? (With apologies to the Christopher Guest film "A Mighty Wind" where we found that made-up word.) Regardless, we use the Triad for our purposes in this paper as it relates to data integration.

[3] information island, "Dictionary.com," *The Free On-line Dictionary of Computing*, Denis Howe, http://dictionary.reference.com/browse/information island (accessed: June 29, 2010). In this paper we add a self-proclaimed "improvement" to this definition.

[4] Clifford Coon, "Ensuring Spreadsheet Accuracy in the Sarbanes-Oxley Era," *IT Audit* 8 (July 1, 2005). In this article, Mr. Coon states that "auditors must monitor whether or not organizations are using adequate internal controls" in relation to spreadsheet management. Mr. Coon touches on many of the same points we do for the rise of spreadsheets (such as ease of use and widespread adoption) and points out that spreadsheet errors can and do cost significant dollar amounts.

[5] Interview with Vice President of Business Development, July 2, 2010.

[6] The Army Portfolio Management System (APMS) has four major modules and manages the Army's portfolio of production systems. Source: http://elamb.org/enterprise-mission-assurance-support-service-emass/ (accessed: July 13, 2010).

[7] Immanuel Kant, *Critique of Pure Reason*, trans. Norman Kemp Smith (New York: Random House, 1958), pg. 48. We make a play upon words here; Kant regarded *space* and *time* as being two *a priori* principles (principles which cannot be learned through experience but must be intuited in the same way that Plato has Socrates elicit "innate knowledge" on geometry from Meno's slave in the dialog of the same name). In this paper, we submit that errors in standalone spreadsheets tend to propagate through time and eventually become almost impossible to discover and correct.

[8] Simson Garfinkel, "History's Worst Software Bugs," *Wired.com*, November 8, 2005, http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all (accessed: July 13, 2010). In this article, we see that on July 28, 1962, the Mariner I space rocket veered off course due to a programming error and had to be destroyed over the Atlantic Ocean.

[9] K. Birman and T. Joseph, "Exploiting virtual synchrony in distributed systems," *ACM SIGOPs Operating Systems Review* 21, no. 5 (November, 1987), pp. 123-138. This paper contains one of the first publicly described publish / subscribe ("pub / sub") models for the "news" subsystem of the Isis Toolkit.

[10] Paul Levinson, "Chapter 10 – Remedial Media," *The Soft Edge: A Natural History and Future of the Information Revolution* (New York: Routledge, 1997), pg. 111. In this chapter, Mr. Levinson takes the reader through the evolution of (surprisingly) *window shades*; that is, human-kind used technology to create walls, then windows to remediate the loss of external awareness that walls brought (first-order remediation), then window shades to remediate the privacy concerns brought about by windows (second-order remediation). Mr. Levinson's point is that technology is uniquely capable of itself providing solutions to the very problems that it brings about.

[11] Zhengxiang Zhu, Wuqi Song, and Jifa Gu, "A Multi-agent and Data Mining Model for TCM Cases Knowledge Discovery," in *2008 ISECS International Colloquium on Computing, Communication, Control, and Management* (Washington, DC: IEEE Computer Society, 2008), pp. 341-346. In this paper, the authors identify how to extract heterogeneous data from Targeted Case Management histories (using Chinese traditional medicine case histories as the target use case). The techniques outlined in the paper apply to most data-mining approaches: a Manager agent to control operations, a Miner agent to scour for data using a set of rules, and a Cooperator agent to perform data correlation.

[12] Brin Sergey and Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Stanford University InfoLab*, Computer Science Department, Stanford University, http://infolab.stanford.edu/~backrub/google.html (accessed: June 30, 2010). This fascinating paper by the co-founders of Google provides their initial Web crawler design (good for up to 100 million pages, which they have far surpassed). Apparently, they had to beg and borrow resources to build their system as graduate students, and the problem of integrity intimately concerned them: "only include the very best documents" in search results.

[13] H. Chris Tseng, "Internet Applications with Fuzzy Logic and Neural Networks: A Survey," *Journal of Engineering, Computing, and Architecture* 1, no. 2 (2007). This article is available online from the publisher at http://www.scientificjournals.org/journals2007/articles/1237.pdf (accessed: July 1, 2010). In this paper, Dr. Tseng discusses how neural networks can use semantics and "fuzzy rules" to track users and make decisions about their interests (basically, to create an ontology around the user searches).

[14] Shon Harris, "Information Security and Risk Management," *CISSP All-in-One (AIO)*, 4th ed., (New York: McGraw-Hill, 2007), pp. 60-61. Ms. Harris, of course, is always ready with a quip and a definition.

[15] Bosworth et. al., *CSH,* "Chapter 9.2.1: Mathematical Models of Computer Security," *Computer Security Handbook: Volume 1*, 4th ed. (Hoboken, NJ: John Wiley & Sons, Inc., 2009), pg. 279.

[16] Ibid, pp. 282-285. Section 9.3 in the text discusses the various types of access control models and the relative merits of each depending on the organization's needs. For example, in the military where data access must be

driven by a user's clearance (trust level) and need-to-know compared to a data item's classification and tags, then a *mandatory labeled* security model provides the best fit.

[17] Charles P. Pfleeger and Shari Lawrence Pfleeger, "Chapter 7: Security in Networks," *Security in Computing*, 3rd ed. (Upper Saddle River, NJ: Prentice Hall, 2003), pg. 373. The seven layers of the TCP/IP model are actually defined in the International Organization of Standards' source document *ISO Publication 7498-2: 1989*. While quoting directly from this source document was a consummation devoutly to be wished, cost precluded that felicitous end. It wasn't free...(as in "free beer" as opposed to "free speech" [with apologies to Richard Stallman and *Copyleft*]).

[18] R. Krikorian and N Gershenfeld, "Internet 0 – inter-device internetworking,"*BT Technology Journal* 22 no. 4 (October, 2004). In this article, the authors decry the current networking requirements that involve expensive computing devices like desktops, specialized network cards to connect the devices to the network, specialized system administrators to configure the machines and the network, and so on. Instead, they define I0 (Internet zero), which postulates seven basic tenets: 1) all devices support the Internet Protocol (OSI Layer 3); 2) such support fits in tiny processors (computational easiness); 3) devices may communicate directly without middlemen (no gateways); 4) devices may "advertise" themselves to become known parts of the larger network environment; 5) *slowing down networks* to decrease complexity; 6) standardizing communications modulation (packaging of message signals for transmission); and, 7) pushing hard for open standards. The project is not really active anymore (see http://cba.mit.edu/projects/I0/) but the idea itself is quite compelling!