Communications of the Association for Information Systems

Volume 14 Article 13

September 2004

Managing Metadata in Data Warehouses: Pitfalls and Possibilities

G. Shankaranarayanan
Boston University, gshankar@bu.edu

Adir Even
Boston University, adir@bu.edu

Recommended Citation

Shankaranarayanan, G. and Even, Adir (2004) "Managing Metadata in Data Warehouses: Pitfalls and Possibilities," *Communications of the Association for Information Systems*: Vol. 14 , Article 13.
DOI: 10.17705/1CAIS.01413
Available at: https://aisel.aisnet.org/cais/vol14/iss1/13

Communications of the Association for Information Systems

MANAGING METADATA IN DATA WAREHOUSES: PITFALLS AND POSSIBILITIES

G. Shankaranarayanan
Adir Even
Boston University
School of Management
gshankar@bu.edu

ABSTRACT

This paper motivates a comprehensive academic study of metadata and the roles that metadata plays in organizational information systems. While the benefits of metadata and challenges in implementing metadata solutions are widely addressed in practitioner publications, explicit discussion of metadata in academic literature is rare. Metadata, when discussed, is perceived primarily as a technology solution. Integrated management of metadata and its business value are not well addressed. This paper discusses both the benefits offered by and the challenges associated with integrating metadata. It also describes solutions for addressing some of these challenges. The inherent complexity of an integrated metadata repository is demonstrated by reviewing the metadata functionality required in a data warehouse: a decision support environment where its importance is acknowledged. Comparing this required functionality with metadata management functionalities offered by data warehousing software products identifies crucial gaps. Based on these analyses, topics for further research on metadata are proposed.

Keywords: metadata, information systems, data warehousing, data quality, business intelligence, ETL, semantic layer, DSS

INTRODUCTION

Metadata is data that describes other data. It is a higher-level abstraction of data that exists within repositories, applications, systems, and organizations. Industry experts believe that metadata offers many advantages for managing complex decision environments such as data warehouses [Kimball et al. 1998]. Reality is that organizations are unable to benefit fully from metadata implementations. Without appropriate controls, metadata evolves inconsistently across the enterprise resulting in pockets of complex, isolated, undocumented, and non-reusable metadata components tightly coupled with individual applications and systems.

A decision environment where metadata is a key factor for success is the data warehouse (DW). The DW is a repository of cleansed, formatted, and well-integrated data that supports reporting and analytical decision-making. The volume of data and the complex data manipulation in a warehouse demands sophisticated back-end processing. On the end-user side, the data serves multiple decision-making needs and must be secure, of high quality, and accessed speedily.

Managing metadata is essential for managing and maintaining the data warehouse. Metadata addresses many aspects of the data warehouse functionality such as data dictionary, process mapping, and security administration. Leading data warehousing software vendors now offer metadata solutions embedded within their products and other vendors offer dedicated software packages for metadata management.

Together with the growing interest in metadata, frustration is growing with attempts to implement metadata solutions. Challenges are not only technical issues, but also cultural and financial ones. The software market does not address metadata needs completely:

- no one product is sufficient to meet all the requirements for managing metadata,
- the lack of accepted metadata standards makes it difficult to integrate metadata across products, and
- metadata implementations are costly, and time-consuming. The end results are often not satisfactory.

Because there are so many facets to metadata and it is interpreted differently by different groups of users, it is difficult to obtain agreement on the design issues and the overall metadata solution. Metadata systems are not cheap, especially when factoring in the effort and time required to gather the metadata requirements and to manage it. Facing such challenges, organizations may choose to build a warehouse with minimal metadata. Such implementations typically restrict the data warehouse functionality and hinder flexibility and ability to expand.

THE LITERATURE

Practitioners discuss metadata from a variety of perspectives. Academic research on metadata is limited and deals primarily with the technological aspects of metadata. Cabibbo and Torlone [2001] introduce a DW modeling approach that adds a new layer of metadata to the traditional DW model. The new “logical” layer is shown to add implementation flexibility and make end-user applications more independent from the underlying storage. Vaduva and Vetterli [2001] review existing metadata implementations and highlight the divergence of metadata exchange standards[1] as a major obstacle for enterprise metadata implementation. Few address the value of metadata for business decision-making and its implications for management. Jarke et al. [2000] propose a data quality meta-model for understanding data quality in warehouses. Chengalur-Smith et al. [1999] and Fisher et al. [2003] show that quality-related metadata does influence decision outcomes. In another study, Shankaranarayanan and Watts [2003] examine the impact of process metadata on the believability of information sources.

SCOPE AND OUTLINE OF THIS PAPER

The growing interest in metadata calls for more research about metadata. The purpose of this paper is to lay out the foundations for such research focusing on metadata in a data warehouse. Section 2 discusses the importance of metadata within the context of a data warehouse environment. It highlights the different roles of metadata in a data warehouse and offers insights into understanding its complexity. Section 3 discusses the challenges involved with implementing a metadata solution including problems with clearly defining metadata requirements, difficulties with financial justification of metadata projects, and inherent technical complexities. Section 4 describes implementation alternatives focusing on metadata management capabilities within commercial off-the-shelf (COTS) data warehousing products, software packages dedicated to metadata management, and homegrown implementations. It discusses the pros and cons of each

[1] Metadata standards such as OIM and CWM are discussed further in Section IV.

alternative besides addressing the functionality requirements and the inherent challenges. Section 5 suggests a set of research issues that calls for an in-depth examination of metadata including directions for evaluating its business value, its role in decision-support, and its ties to knowledge management.

II. METADATA IN DATA WAREHOUSES

To illustrate the role of metadata in decision environments, we examine its roles in the data warehouse, an environment where its necessity is widely acknowledged. Figure 1 presents a conceptual architecture[2] of a data warehouse environment.

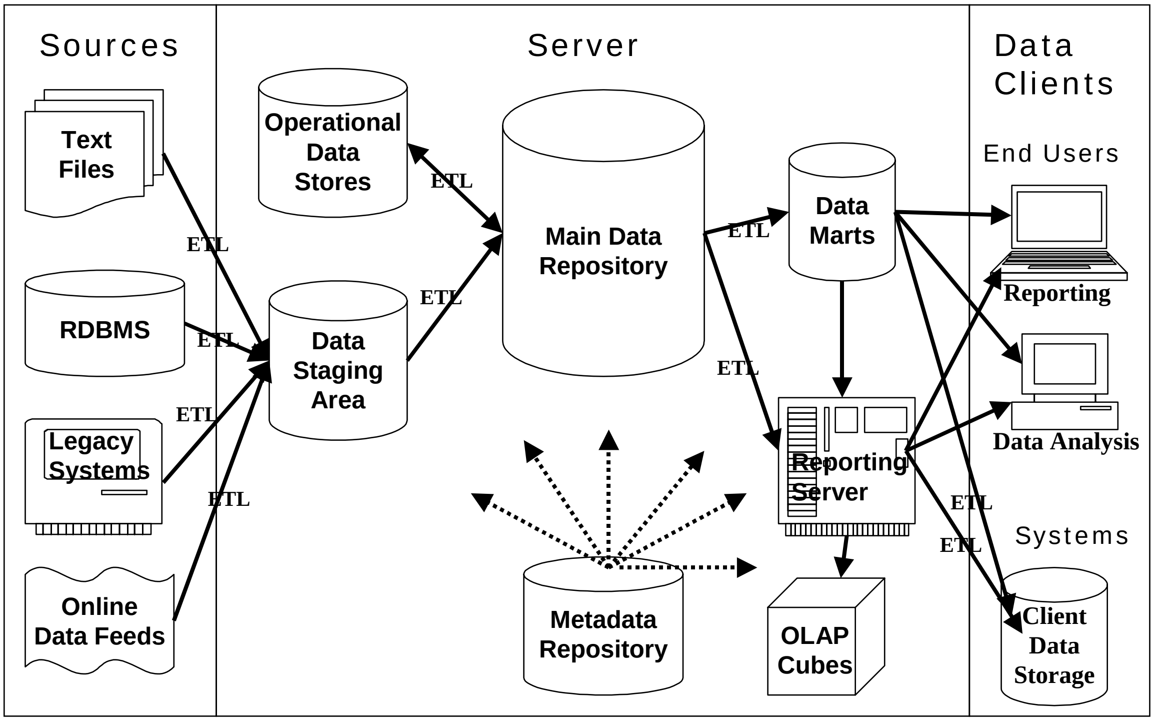

Figure 1. Data Warehouse Architecture

DATA WAREHOUSE COMPONENTS

The set of components in a typical data warehouse includes data sources, warehouse server, ETL processes, and data clients.

Data Sources

A data warehouse typically contains multiple operational data sources, internal and external to the organization. Sources of data may be text files (in various formats such as ASCII-delimited, MS-Excel, HTML, and XML), relational database management systems (such as Oracle, MS-SQL, and Sybase), legacy systems (mainframe-based files using VSAM or ISAM) and online data feeds (such as stock quotes or sensor outputs submitted by application interfaces).

[2] In practice, other variants of architecture and terminology exist.

Warehouse Server

The core of a data warehouse is the set of components that manipulate, load, and store data and serve data to users of the data warehouse. The warehouse server is a term that refers to this collection of components and typically includes:

(1) a primary data repository in which the permanent warehouse data is stored,

(2) a data staging area, a temporary storage where data is collected from different sources, cleansed, reformatted, integrated and aggregated before it is loaded to the primary repository,

(3) an operational data store (ODS) where subsets of warehoused data can be manipulated independently,

(4) data marts – customized subsets of data that serve specific departmental needs,

(5) a reporting server that makes data available for end-user applications, either in two-tier client-server configuration or three-tier configuration for accessing reports via the Intranet / Internet. To improve performance, some reporting servers reformat the data and store it in local, internal data structures such as OLAP (On-Line Analysis Processing) cubes.

(6) the metadata repository where metadata is stored.

ETL (Extraction-Transformation-Loading) Processes

Back-end data processing systems in the warehouse work in three stages:

1. Extraction that pulls the data out of the data sources,
2. Transformation that converts the data into a desired format (may include processes such as data cleansing, error correction, integrating multiple data sources, restructuring data and aggregation), and
3. Loading that moves the transformed data it into the desired location in the warehouse.

Data Clients

Warehouse data can be provided either directly to end-users or indirectly through other systems. Given the complex end-user requirements, delivering data to end-users is a programming-intensive task that typically uses sophisticated off-the-shelf software packages, including reporting and BI (business intelligence) tools.

THE IMPORTANCE OF METADATA

Metadata is necessary for managing and administering a corporate data warehouse. Metadata is also beneficial to the users (decision-makers) who consume data from the warehouse [Fisher et al. 2003]. The importance of metadata may be illustrated from three perspectives: data management, knowledge management, and data quality management.

Data Management

Metadata helps in managing the data and making better use of it. Organizing and “cataloging” the data allows for more efficient searching, on-going maintenance, integrity preservation, and control over data redundancy. The abstraction permits the data to be managed independently from applications and systems that access it. Metadata also acts as an intermediate layer between business users and the information systems and applications that they have access to. This layer permits increased flexibility in managing customers, their changing requirements, and their use of resources within the organization. The metadata repository may offer insights to “often-used” versus “dormant” data by tracking usage patterns. Using this information, it becomes easier to anticipate changes in user requirements and take proactive measures when identifying new information product offerings.

Knowledge Management

By providing an abstraction of business and technical information, a metadata repository promotes sharing and reuse of knowledge, expertise, and application components. The metadata repository makes it easier to setup, maintain, and grow systems by explicitly documenting the in-depth knowledge about systems and all their components in a format that can be easily retrieved and interpreted. The metadata thereby facilitates system maintenance and reduces the time and effort spent in system upkeep. Further, when new systems and applications are being built, a considerable amount of this captured knowledge can be reused. As a result, development effort is reduced and the time-to-deliver of new applications is shortened. The metadata also captures the “meta-information” about each application and its modules. Metadata permits “mass-customization” of applications. It eliminates the development of each application (or deliverable) as a “one-off” by allowing developers to leverage and reuse the metadata on existing applications when developing a new one.

Data Quality Management

Metadata can be used to evaluate data quality in decision environments. Data quality is becoming an important issue in information systems. In organizations, capital losses and heightened risk exposure are increasingly being attributed to data quality issues such as accuracy, consistency, completeness, and timeliness. For many systems, these issues are becoming so common that system usability is affected. Managing and improving data quality requires an understanding of the data flow through the information system and continuous monitoring of quality throughout. Both require the support of a metadata layer. The metadata captures the complex information within a warehouse such as how data was obtained, what business rules were applied to it, how it was transformed, and what constraints are associated with it. This information can be communicated to the user on-demand, which improves the perceived quality of the data and the decision environment and thus increases the user’s confidence in the environment.

METADATA FUNCTIONALITY

An important step to implement a metadata solution successfully is to understand its functional requirements. Traditionally, IS professionals viewed metadata as the system’s data dictionary that captures the definitions of data entities and their inter relationships. This narrow view of metadata ignores its richness and hides its complexity. In recent years the view of metadata has broadened to include other components that typically exist in information systems. Few research papers examine metadata taxonomy and propose methodologies for metadata classification in a manner that helps recognize the complex nature of metadata. Functionality is an important dimension for such a classification. Metadata serves many different functions in an information system, especially in decision support environments. To understand the usefulness and its complexity the different functionalities that it serves are described in Sidebar 1.

The following paragraphs are a taxonomy that classifies metadata in a more granular/specific fashion and extends the three classifications discussed in Sidebar 1. The taxonomy provides a comprehensive list of metadata elements, classified based on metadata functionality. Within each type, the metadata elements are further classified, using the other suggested perspectives. This taxonomy offers an enhanced and a more modular view of metadata that improves the understanding and gathering of metadata requirements. It includes data dictionary metadata, data delivery metadata, process metadata, quality metadata, infrastructure metadata, and administration metadata.

SIDEBAR 1. METADATA FUNCTIONALITIES

Kimball [1998] classifies metadata as back-end and front-end depending on whether the metadata is associated with back-end processing (such as extraction, transformation, loading) or with front-end data delivery (such as end-user preferences, reporting tools, user management, and administration).

Marco [1998] classifies metadata as business and technical. Technical metadata includes abstractions needed for the system operation. It targets IT developers and warehouse administrators. Business metadata targets the users of the data warehouse and includes information essential for effectively using the data, applications, and systems.

Imhoff et al. [2003a] classifies metadata as business, technical, and administrative. The administrative metadata includes metadata for managing the performance of a warehouse such as audit trails, performance metrics, and quality metadata besides the metadata for administering the users (security and access control).

Data Dictionary Metadata

Data dictionary metadata (Table 1) includes definitions of the data entities being maintained and the relationships among them. The data dictionary captures storage information at different levels such as data repositories, databases, tables, and fields. It also includes (1) the semantic layer necessary to translate the data elements in the source databases to those in the data warehouse, and (2) the business terms necessary for end-users to interpret the data in the warehouse.

If metadata is used as a solution for data integration within the data warehouse, the data dictionary includes the data modules necessary to map the “vocabulary” across multiple user-groups or business units.

Table 1. Data Dictionary Metadata

	Business	**Technical**
Back-End	• Business interpretation of data items	• Data structure, which depends on the storage type - text files, spreadsheets, RDBMS, Legacy systems, data streams, Lotus Notes database or others. • Elements specific to RDBMS storage – tables, fields, indexes, views, stored procedures, triggers • Mapping data elements between sources and data warehouse
Front-End	• "Semantic Layer" of naming and definitions of data items in "business" language. • Documentation, on-line help and training aids • Reports contents and formatting	• Mapping data items to database tables and fields or file locations • Data extraction or manipulation syntax behind GUI visualization. • Default formatting preferences • Syntax for joining multiple data sources • Mapping of data elements between warehouse and user-applications.

Data Delivery Metadata

Data Delivery Metadata (Table 2) is also known as reporting metadata and is front-end metadata. It includes metadata on report templates used for delivering data from the warehouse, where fields are linked to one or more data warehouse elements. Data warehouse reports typically follow a hierarchical order in which the dimensional fields are organized (e.g. time may be hierarchically organized as days, weeks, months, and quarters). Dimension hierarchies may also be associated with reports. This metadata further includes template files (Cascade Style Sheets or CSS for web outputs, or Formula-1 templates for Excel-like outputs, for example) used for displaying reports. Data delivery metadata defines the reports that constitute a dashboard and their presentation order.

Table 2. Data Delivery Metadata

	Business	Technical
Back-End	N/A	N/A
Front-End	• User vocabulary • Metaphors for data visualization • Personalized aggregation and other computation definitions • User defined dimension hierarchies	• Metadata on report templates including report fields and layouts • Physical location of template files • Mapping of report fields to warehouse data elements • Format preferences for data elements • Mapping multiple reports into consolidated dashboards

Process Metadata

Process metadata (Table 3) captures information on how the data stored in the warehouse was generated. This component of metadata corresponds primarily to ETL processes (see earlier under Warehouse Components), describing how data was transferred from sources to targets and what manipulations were applied during transfer. Process metadata serves both technical and business users. Administrators use process metadata for activating and managing the ETL processes. Business decision makers can use it for better data quality assessment based on their experience with the data sources and/or their understanding of data manipulations applied. Shankaranarayanan et al. [2003] propose the Information Product map (IPMAP) as a technique for representing the processes in the manufacture of an information product. Outputs of information systems can be treated as information products [Wang et al., 1998]. This technique can be adapted for a data warehouse. The warehouse data can be treated as an information product and its manufacture represented by the IPMAP.

Table 3. Process Metadata

	Business	Technical
Back-End	• Business interpretation of the transfer processes, including and business rules for: • Source-Target mapping • Newly created fields • Source integration • Aggregations • Filtering • Tracking versions • Process charting (IPMAP)	• ETL engines • Technical implementation of the data processing business rules • Technical configuration of data staging • Data transfer schedule, contingencies, tracking and synchronization • Data extraction software or API's provided by the source • Schema and format adjustment between source and target • Monitoring system resource usage during transfer
Front-End	N/A	N/A

Quality Metadata

Quality metadata (Table 4) helps evaluate the quality of the data in the warehouse. Quality may be evaluated in an “impartial” or “objective” manner without considering any contextual factors. Alternately, decision-makers may evaluate quality based on contextual factors such as the decision-task and the decision-maker’s expertise. To support both objective and contextual evaluation, quality metadata includes not only evaluated measurements of data quality but also metadata necessary for decision-makers to assess the quality of data. Data quality is evaluated along dimensions such as accuracy, completeness, timeliness, and consistency. Such dimensions can be evaluated using the IP approach [Ballou et al. 1998, Shankaranarayanan et al., 2003]. The IPMAP representation that is used to capture and communicate the process metadata can be supplemented with the metadata necessary to assess data quality.

Table 4. Quality Metadata

	Business	**Technical**
Back-End	• Definition of quality measurements • Data quality measurement values • Business rules for data inspection and cleansing	• Utilities for automated data quality assessment • Automated data cleansing • Utilities for data inspection
Front-End	• Numeric presentation or graphical visualization of quality measurement • Quality dimensions & reporting	• Making quality metadata available for reporting and data analysis utilities.

Infrastructure Metadata

Infrastructure Metadata (Table 5) contains information on system components and abstracts the infrastructure of the information system. It is used primarily for system maintenance and improvements.

Table 5. Infrastructure Metadata

	Business	**Technical**
Back-End	• Business identification of systems • URL's	• Maintenance of hardware, OS, Database servers and network protocols • Network addresses • Tracking use of hardware and software • Database administration, including • Server configuration – databases, partitioning and administration scripts • Performance optimization, Capacity and utilization • Database backup procedures
Front-End	Similar to the above	• Maintenance of hardware, OS, Database servers and network protocols • Network addresses

Administration Metadata

Administrative metadata (Table 6) includes metadata necessary for managing the security of and access privileges to data in the warehouse and applications associated with it. This metadata may include information on how report templates are shared by multiple users and metadata necessary to restrict the data elements that are valid for each template field depending on the user.

III. CHALLENGES WITH METADATA IMPLEMENTATION

IT practitioners face significant challenges when implementing metadata solutions. A major challenge is the understanding and the gathering of metadata requirements. As described in Section II, metadata is inherently complex and multi-functional. The abstracted and technical nature of various metadata elements makes it hard to introduce metadata concepts to business people, obtain their feedback, and reach an organization-wide consensus about requirements. Moreover, the lack of robust models for evaluating metadata makes metadata investments difficult to justify economically. Significant challenges are introduced by the many available architecture and implementation alternatives. Choosing the right alternative may determine the success of the metadata initiative. This idea is discussed in greater detail in the ensuing sections. Other challenges stem from technical complexities associated with metadata implementations.

Table 6. Administration Metadata

	Business	**Technical**
Back-End	• System Ownership • Usage privileges, usernames and passwords • Groups and Roles for business functions • Legal limitation on data use for both sources and targets • Tracking use of data items	• Authentication interfaces • Application and Data security
Front-End	• Users and passwords • Data and tool use privileges • Delivery configuration • Tracking data and tool usage • Report template/layout • Template visibility and sharing	• Tracking report delivery failures • Personalizing data delivery formats and styles including metaphors for visualization • Valid data elements in each field of the template layout

Such implementations require the use of different representation formats and significant integration efforts to create a uniform representation.

METADATA FORMATS

Metadata can be stored in data warehouses in several different formats. A summary of alternative formats for representing metadata is shown in Table 7. The table also describes the advantages and disadvantages associated with each format.

The need for multiple formats increases implementation complexity, particularly when consistency is needed among formats that are different in nature. A common approach for addressing this issue is to choose a master format as a baseline and map all others to it. Typically, the relational model is chosen as the master since it is highly structured, easily accessible, and well supported by software tools. The document format is harder to integrate. Documents are typically less structured and hence might require significant efforts for capturing the metadata stored within and integrating it with other formats. When such need arises, Information retrieval techniques are applied to extract key words and concepts from such documents that are then captured in relational databases. This task is time intensive. Several iterations are usually needed before useful metadata is extracted from the documents.

Table 7: Data Formats for Metadata

Method	**Description**	**Typical Components**	**Pros**	**Cons**
Text Files	Configuration files in shared directory. Information is obtained through file parsing	• Hardware and OS configuration	• Low cost • Readable • Easy to manipulate	• Hard to centralize • Less secured • Hard to capture complexity • Requires file parsing software
Relational Models	Database model implemented on RDBMS server. Access through native or ODBC drivers	• System tables in RDBMS • ETL process configuration • Semantic Layer • Security configuration	• Easier to centralize • Open architecture • Standard access method • Database administration	• Relational modeling may turn too complex • Database administration overhead • Dependency on RDBMS server

Method	**Description**	**Typical Components**	**Pros**	**Cons**
			utilities	platform
Hidden structure	Metadata is stored in internal data structures. Read/Write access via dedicated API	Similar to Relational Modeling, vendor dependent	• Data is better secured • Consistent access • More efficient for application use	• Access depends on API, Data is unreadable otherwise • May prevent integration with other tools
Graphics	Graphical chart or diagram format	• ER Diagram • IPMAP • Architecture chart • Monitoring utilities	• Allow easier absorption of large volumes of information	• Hard to integrate with other components • Training and interpretation skills
Documents	Textual documentation – documents, spreadsheets, HTML	• System documentation • Semantic layer • Training materials	• Readable by business users • More flexible with capturing complexities	• Hard to integrate with technical layers

METADATA INTEGRATION

The evolution of isolated islands of metadata is typical in a data warehouse. Each island is made up of a specific type of metadata and tailored to a specific sub-system of the warehouse. Many IT practitioners believe the lack of integration might negatively impact the ability to grow and maintain the warehouse in the long run. The benefits of integration can be illustrated from the following perspectives:

Data Standardization

Data entities often lack standardization among different components of the data warehouse: data storage, ETL, and reporting. This lack raises data integrity and data consistency issues in the warehouse. It is not uncommon for different business units to interpret the same business entity (e.g. customer) or formats for storing date/currency values differently. Maintaining coherence and integrity requires integration of metadata where the formats and definitions of business entities are captured.

System Integration

The data warehouse environment requires significant integration among systems at the data level. Data is transferred from sources system into the warehouse, stages into a different format, stored in a repository, and delivered to end-user and client systems. Efficient data flow among systems requires them to be able to communicate and understand each other. Integrated metadata is essential for ensuring seamless systems integration.

User Administration

In typical implementations, user access and security is managed using different utilities or tools and at different levels (system, database, and application to name a few). Firms that use a single sign-on (where authentication and authorization are managed centrally across all levels of the information infrastructure) require integration of the administrative metadata across all subsystems of a data warehouse.

Metadata integration in a data warehouse can also be examined from business, logical, and physical perspectives proposed by Jarke at al. [2000]. The business perspective puts the enterprise model at the center and focuses on addressing business needs. The logical perspective looks at the data warehouse model with its data entities and focuses on the logic behind how data is generated, stored and used. The physical perspective looks at the actual hardware and software that support the warehouse. Metadata components are primarily designed to address one perspective or the other and do not easily integrate across the three.

To create an integrated solution, an organization may implement a centralized metadata repository [Sachdeva, 1998; White, 1999]. This approach puts all the metadata components into one unified repository. A key challenge with implementing an enterprise metadata repository is to develop a universally (enterprise-wide) accepted standard. The development of such repositories raises significant issues in terms of design paradigms, system architectures, and centralization vs. de-centralization. These issues are discussed next.

III. METADATA IMPLEMENTATION METHODOLOGIES

Organizations recognize the need for integrated metadata. When faced with the challenges described in Section III, organizations sometimes resign to adopting the “do-nothing” approach for managing metadata. While the “do-nothing” approach does not require any investment, it results in increased time and effort to manage the warehouse. Opportunities for leveraging and reusing knowledge and expertise are lost. As a consequence, more metadata islands are created, compounding the existing problems. We describe three other alternatives for metadata management: – leveraging metadata management capabilities offered by commercial off-the-shelf data warehousing tools, adopting commercial products dedicated for centralized metadata management, or implementing home-grown metadata solutions. These alternatives are summarized in Table 8.

Table 8. Metadata Implementation Methodologies

Methodology	Description	Pros	Cons
“Do Nothing”	Making no significant attempt to implement a metadata layer or integrate existing ones	• Minimal investment in development efforts • Granting each system and sub-system with full implementation flexibility	• “Information Chaos” • Lack of standardization • Loosing the ability to integrate systems • No leveraging and reuse of knowledge • Increasing on-going maintenance overhead
Data Warehousing Tools	Using metadata management capabilities within commercial off-the –shelf (COTS) data warehousing tools	• Metadata is already embedded within the tools. No additional licensing cost • Some tool suits provide comprehensive metadata capabilities • Shortening the development cycle	• Metadata focuses on the specific tool needs. • No single tool supports the entire set of metadata functionality requirements • DW Tools tend to focus on the technical aspects of metadata and provide minimal or no support for business aspects • Metadata integration and exchange between different tools is not well-supported
Metadata Management Systems	Using software packages that specialize in metadata management	• Promotes metadata centralization and integration • Support a wider range of metadata functionality and formats. • Addressing business aspects, not only technical ones	• Licensing overhead • Development and training efforts • Most packages are new to the market and had not matured yet
Home-grown Implementation	Self-development of a metadata layer	• Customizing to organizational needs • Better control over contents and functionality • Saving on licensing costs	• “Reinventing the wheel” • Significant programming efforts • Longer implementation cycles

METADATA MANAGEMENT USING DATA WAREHOUSING TOOLS

Since the early 90's, many of the 3rdparty data warehousing tools offer utilities for metadata management. A review of the leading COTS products shows four key drawbacks of using such tools.

1. None offers a comprehensive set of capabilities needed to manage all types of metadata (even excluding quality metadata). Hence none can manage an integrated metadata repository.

2. The products emphasize technical metadata while business metadata is poorly supported or ignored.

3. Storage and presentation formats of metadata are restricted to relational and proprietary structures. This limits implementation flexibility.

4. The metadata elements are tightly coupled with one tool or a suite of tools (by the same vendor) and little is offered by way of common interfaces for metadata exchange. Metadata integration across tools is thus difficult.

The ability to create a well-integrated metadata layer is needed if the data warehouse is to be successful [Marco, 1998], [Inmon, 2000]. Practitioners identify the metadata integration capabilities of COTS products as an important factor in the success of metadata implementations [Sachdeva, 1998, Seeley and Vaughan, 1999]. Software vendors such as IBM, Microsoft, and Oracle are aware of this problem and offer reasonable internal integration within their offerings. The Meta Data Coalition (MDC), established in the late 90's by some of the key firms in the data warehousing market, was the first major force for the standardization of metadata exchange. MDC proposed the Open Information Model (OIM), a data exchange protocol that permits each tool to maintain its own internal metadata structure and offers a uniform interface for metadata exchange. OIM is the standard adopted by Microsoft. The other standard is the Common Warehouse Model (CWM). CWM was proposed by the Object Management Group (OMG), which includes vendors such as Oracle, Hyperion, IBM, NCR, Informatica, and Unisys. CWM is based on XML metadata interchange (XMI), the data interchange standard for distributed applications. The existence of two competing standards might be a major obstacle for integration efforts. Seeley and Vaughn [1999] offer insights into the struggle for defining the metadata standard including the underlying politics.

Another problem with using data warehousing tools for integrating metadata is the narrow view of metadata adopted by these tools. Most tools emphasize data dictionary metadata, and integration capabilities, if offered, are for integrating data dictionary elements only. For products in which other types of metadata can be captured and managed (process metadata with ETL tools, administration and data delivery metadata with reporting and business-intelligence tools), the metadata is captured in proprietary structures that preclude integration of metadata repositories.

Table 9 summarizes the metadata management capabilities of commercial data warehousing tools. The tools are grouped into four categories – data storage (RDBMS), On-Line-Analysis-Process (OLAP) servers, ETL, and business intelligence (BI).

The offerings of the various vendors are discussed in Appendix I. The vendor products are grouped into data storage, ETL, reporting/business intelligence, and metadata management products. Homegrown metadata repositories are discussed next.

Table 9: Metadata Support in Data Warehousing Products

Main Product Offering	Company	Metadata-Integrated Offerings[3]	Format[4]	Functionality Focus[5]	Main Exchange Standard
RDMBS	Oracle	O,E,B,D	R	DIC	CWM
	Microsoft, MS-SQL	O,E,B	R	DIC	OIM
	Sybase	O, D	R	DIC	CWM
	IBM, UDB	O,E,B,D	R	DIC	CWM
	Teradata		R	DIC	CWM
OLAP	Hyperion		P	DIC	CWM
ETL	Informatica	B	R	PR	CWM
	Hummingbird		R	PR	None
BI	Cognos	O, E	P	DEL	CWM
	Business Objects	O, E	R, P	DEL	CWM
	MicroStrategy		R	DEL	None

HOMEGROWN METADATA REPOSITORIES

Homegrown implementations were the only option for metadata management in the early days of data warehousing and are still favored by many organizations. The Dublin Core project (http://www.dublincore.org), for example, is a consortium of organizations that adopted the homegrown approach. Dublin Core is attempting to standardize metadata and support implementation efforts[6]. Homegrown metadata implementations vary from simple text-file or spreadsheets to sophisticated web-based solutions. It allows better control over the metadata components and functionality. The simple implementations tend to create metadata islands that reside on personal computers or are tied to one application making integration/sharing difficult. The sophisticated approaches are more complex and risk exceeding time and effort estimates. Our focus here is on the integrated and sharable metadata solutions. Two key issues to address in such projects are the design paradigm and the architecture for the repository.

Design Paradigm

An important consideration in the process of implementing metadata is the design paradigm: top-down, bottom-up, or a compromise between the two. A top-down approach looks at the entire organizational information system schema and tries to capture an overall metadata picture. A bottom-up approach, on the other hand, starts from a low-level granularity of subsystems and bring their metadata specifications together into a unified schema. While a top-down paradigm is better for standardization and integration among sub-systems, it might be infeasible where existing information systems with local metadata repositories are already in place. Moreover, capturing organizational metadata requirements is a complex, tedious, and time-consuming task. The bottom-up paradigm is more likely achieve to short-term results, but might fail to satisfy broader integration needs. A compromise alternative is a “middle-out” approach. This approach treats each type of metadata as a module or component of the larger repository. It starts with identifying one or two key (depending on the warehouse and its use) modules of metadata and building a repository consisting of these modules. The functional classification of metadata (described under Metadata Functionality) can help identify such modules. It is important to decide on a format that will facilitate integration and on a metadata exchange model (e.g. Common Warehouse Model). It should also be recognized that each module will not be comprehensive or exhaustive to start with but will grow incrementally over the life of the warehouse. Subsequent to

[3] R-RDBMS, O-OLAP, E-ETL, B-Business Intelligence, D-Design
[4] R-Relational, P-Proprietary
[5] DIC – Data Dictionary, PR – Process, DEL – Data Delivery
[6] See http://www.nsdl.org , as an example for adopting the Dublin Core standards**.**

the implementation of this initial "core" repository, other modules may be added on one-at-a-time to extend it and existing modules may be extended as well. The advantages of this approach from a business standpoint are:

1. it requires minimal investment initially. As the value is recognized, additional investments can be made,
2. it is custom-developed to meet the specific, complex requirements of the organization and its offerings, and
3. it can be designed and built on an existing hardware/software platform and ported to a larger, more sophisticated platform at a later date should the need arise.

This approach will further ensure that the metadata and its repository structure remain extensible and not dependent on a single set of applications. Figure 2 presents a conceptual layered architecture of a homegrown metadata repository illustrating the different modules within. This architecture is targeted for total data quality management by including process and quality metadata.

Reporting metadata is linked with the warehouse data elements for effective communication with decision-makers. Together with user vocabulary preferences personalized metaphors and formatting, it constitutes the data delivery metadata in the warehouse. The mapping component of the data dictionary metadata consisting of the mapping between data elements is distributed across the conceptual layers (shown by the arrows in Figure 2). The data dictionary metadata elements used in data integration such as the dependencies and constraints between source data elements are captured in the middle and lower layers of the conceptual architecture. Data quality metadata is integrated with process metadata and is represented by the IPMAP in the conceptual architecture. These processes are mapped to the extraction and transformation rules (another piece of the process metadata) captured in the middle layer of the architecture. Administration and infrastructure metadata that spans the entire warehouse are shown on either side of the conceptual architecture in Figure 2 on the next page. A detailed list of metadata elements corresponding to the architecture in Figure 2 is presented in Table 10.

Table 10: Metadata Elements for TDQM in a Data Warehouse

Metadata Entity	**Metadata items**
Warehouse Data Elements	Date loaded, Date updated, Currency (old/current) in the warehouse, associated data sources, associated extraction, cleansing, and transformation processes, whether (still) available in the data source, associated staged data elements, staged data sources
Data Sources	ID or Unique name, Format type, Frequency of update, Active Status
Source Data Objects	Object name, Aliases, Business Entity name, Business rules associated, Owner
Source Data Elements	Element name, Units, Business rules, Computation method(s), business name/alias, data type, data length, Range-Max, Range-Min, Date/time when it was included, Constraint and participating source elements
Intermediates /Target Objects	Object name, Aliases, Business Entity name, Business rules associated, Owner, Creation date, Object Status, Administrator,
Intermediate and Target Data Elements	Element name, Units, Business rules, Computation method(s), business name, data type, length and range, Date/time when it was included or became effective, [Constraint and participating source elements]
Source to Target Mappings	Derivation and business rules, assumptions on default and missing values, associations between source and target data elements

ETL Process Modules	ID and Unique name, Creation date, Effective date, Owner, Business Unit responsible, Modification information, system/platform associated, location in file system, execution commands, Run Date, Error messages
Extraction Process	Applicable source data element(s), extraction rules, business restrictions, Last Run Date, Error Codes/Messages, output data elements
Cleansing Process	Applicable source data element(s), sanitizing rules, business restrictions/rules, output data elements
Transformation Process	Input data element(s), transformation rules, business rules, output data elements
Load Process	Input data element(s), format/transformation rules, business rules, output warehouse data elements

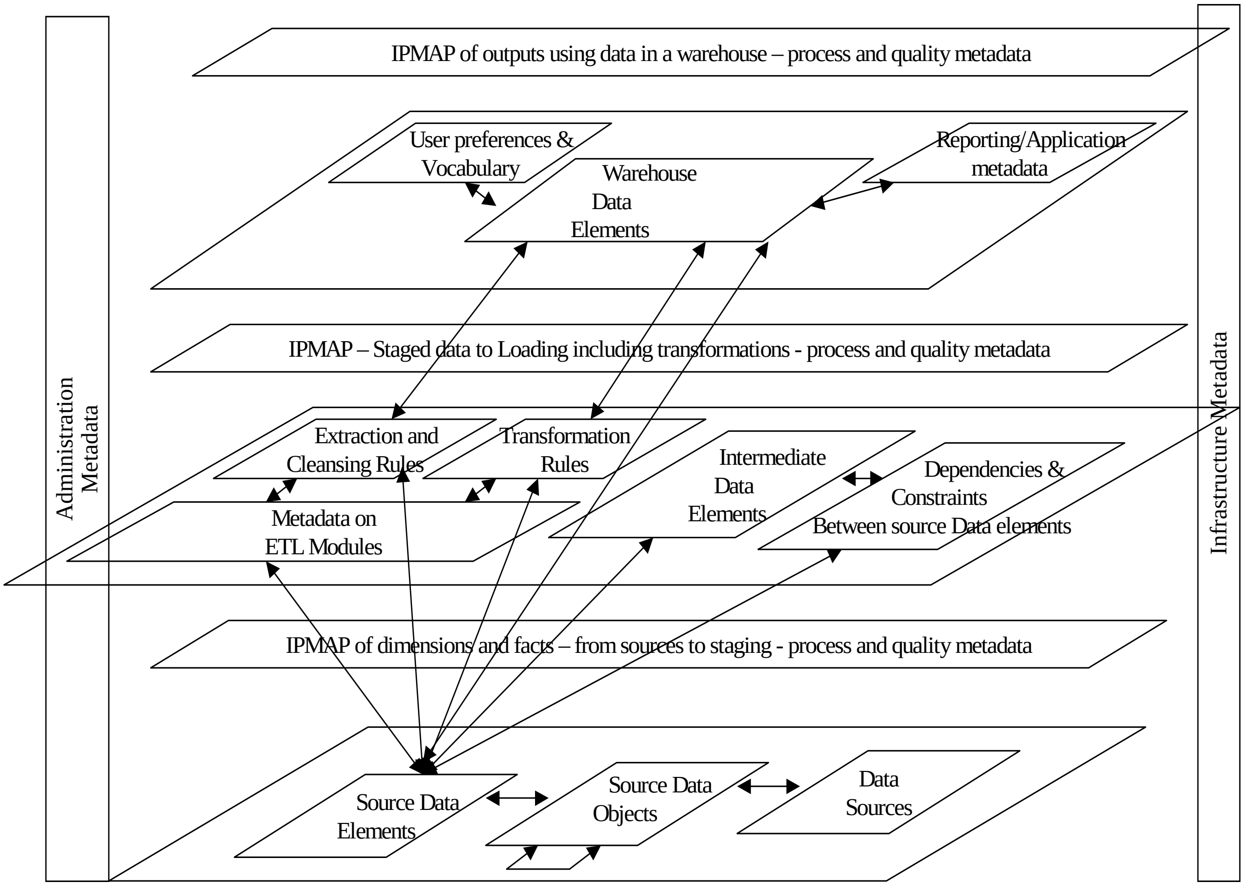
Administration Metadata
Infrastructure Metadata
IPMAP of outputs using data in a warehouse – process and quality metadata
User preferences & Vocabulary
Warehouse Data Elements
Reporting/Application metadata
IPMAP – Staged data to Loading including transformations - process and quality metadata
Extraction and Cleansing Rules
Transformation Rules
Intermediate Data Elements
Dependencies & Constraints Between source Data elements
Metadata on ETL Modules
IPMAP of dimensions and facts – from sources to staging - process and quality metadata
Source Data Elements
Source Data Objects
Data Sources

Figure 2: Conceptual Architecture for Metadata Repository

Metadata Architecture

The choice of centralized versus distributed architecture continues to be a debatable topic in information systems. Mimno [2002] discusses this issue in a data warehouse and suggests a hybrid approach that allows greater flexibility of sub-systems and shortens development cycles. At the same time, it preserves an acceptable level of control. The same debate is also relevant to metadata implementations. Blumstein [2003] describes three alternative metadata architectures (summarized in Table 11): centralized, distributed, and hybrid. The hybrid approach offers stringent control of the metadata that resides in a central repository. It also permits better sharing and reuse of metadata as individual applications maintain local repositories, each with application-specific metadata. This architecture permits quicker response and independence as compared to either a centralized or a distributed architecture. Information exchange between the central and the local repositories is bi-directional using XML metadata exchange standards.

Table 11: Summary of Metadata Repository Architectures

Architecture	Implementation	Pros	Cons
Centralized, (Passive Repository)	Single, centralized location for metadata. All beck-end (Data storage, ETL) and front-end (Business Intelligence) tools should post their metadata into the repository, which becomes the only source for pulling and using it.	• Efficient access - No need to search for Metadata in multiple locations. • Better performance, no need to communicated with multiple tools • Independence from tools being activated or not • Easier to standardize and integrate • Easier to capture metadata not related to a specific tool	• Complex and time consuming implementation • Data redundancy with larger chance for quality hazards • Synchronization issues • Increased maintenance efforts
Distributed (Active Repository)	Metadata is kept on the back-end and front-end tools. The users access a single repository, which doesn't maintain copies but retrieves the metadata in real-time, as needed.	• Efficient access - single location with light-weight data requirements. • Faster application development due to higher level of independence. • No data redundancy, metadata is kept at its source. • Reduced maintenance	• Dependency on the systems being active • Harder to standardize and integrate • Harder to capture additional metadata, not supported by the end-tools.
Hybrid	Pieces of metadata provided by back-end and front-end tools are kept at the tools and accessed in real-time, while home-grown pieces are at the repository.	• Efficient access • Application independence is kept • No data redundancy • Ability to integrate between 3rd-party and home-grown metadata	• Sophisticated to implement • Integration might not be achievable • Dependency on the end-systems being active

Though the hybrid architecture captures the advantages of the two extremes, it is highly sophisticated and might be "overkill" for simple environments. A warehouse implementation should evaluate the following to decide the appropriate architecture:

1. the level of organizational distribution: geographical and departmental,
2. back-end and front-end tools already in place, or planned for,
3. expected data volume and complexity of the data warehouse, and
4. estimated implementation and maintenance efforts.

The design paradigm chosen and the derived architectural approach are highly related and likely to be influenced by the organizational structure and its information system complexity. It is unlikely that a large organization with sophisticated information needs would adopt a top-down design of metadata and implements it in a centralized manner. Having information systems already in place, such organizations are more likely to adopt a decentralized architecture, or a hybrid, using a “middle-out” design paradigm. Smaller organizations, with less complex information demand, can afford the luxury of a top-down approach and attempting to capture the entire set of metadata requirements. These organizations can also adopt a centralized architecture.

A recent trend in data warehousing is the Virtual Data Warehouse (VDW) [Holland 2000, Imhoff 2003b]. Data is never moved into a single warehouse repository but continues to be stored in the source (transaction or legacy databases or even a different warehouse). Using data access components and processes, the separate sources are integrated in a manner that is transparent to end-users. The VDW finds application, for example, in situations where time-dimensioned data stored in a warehouse must be combined with on-line (possibly real-time) transactional data for certain types of analysis. A good example is online stock trading applications. For efficient trading, the trader looks at both historical data of stock prices and online data of current market trends. Such integration demands very sophisticated metadata repository designs and structures that are beyond the scope of this paper.

V. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

CONCLUSIONS

This paper presents a comprehensive examination of the state of metadata in the data warehouse. The purpose of this examination is to emphasize the need for further research on metadata by highlighting two issues:

1. The importance and benefits of good integrated metadata
2. The challenges in successfully implementing integrated metadata repositories.

The taxonomy of metadata that was presented in this paper is based on the functionality that metadata serves in a data warehouse. It illustrates the multiple classes of metadata that support a wide spectrum of functionality in a data warehouse. This taxonomy supplements the existing metadata classification schemes and the paper illustrates how the existing and the proposed classification schemes fit together. The software industry acknowledges the importance of metadata and the difficulties with implementing metadata solutions. The software vendors now offer metadata management capabilities within data warehousing products or as separate software packages[7]. Unfortunately, there is still a mismatch between the capabilities needed for managing metadata and capabilities that these products offer (as seen in Appendix 1). The state of industry offerings highlights the need for a metadata standard to achieve integration and exchange of metadata across repositories. As an alternative to managing metadata using commercial products, the paper discusses the implementation of a “homegrown” integrated metadata repository and the associated challenges.

DIRECTIONS FOR FUTURE RESEARCH

An important observation is the significant difference in how technical and business metadata are perceived. Administrators, IT managers, and other technical users of metadata clearly recognize the merits of metadata. Business users, however, perceive metadata as a technical necessity and do not recognize the implications of metadata, particularly business metadata[8]. This observation raises the question: “To what extent is metadata useful to business users?” There are a few perspectives for addressing this question - better assessing the operational value of metadata, understanding its possible implications for decision-making, and exploring its association with organizational knowledge.

[7] The metadata management capabilities offered by data warehousing tools are summarized in Appendix I.

[8] Business metadata (Section II), includes metadata components that are aimed for business users. For example, the source of a data element, business rules applied to manipulate it, assumptions and models used in the manipulation and other information that helps evaluate the usefulness of that data element for their purposes. However, few business users are trained in using metadata and hence ignore or overlook metadata even if it is made available to them.

Operational Value of Metadata

The benefits of metadata remain largely intangible and there is a lack of models or methodologies to evaluate and quantify its operational value [Stephens, 2004]. Suggestions for looking at metadata-ROI have been offered by practitioners [Marco 2000]. The following issues should be explored to better understand the operational context of metadata as a step towards developing such evaluation models:

1. What functional types of metadata are the most significant operationally?

> The consolidated taxonomy of metadata that was offered (see Metadata Functionality in Section II) can serve as a baseline for exploring this question – are certain (functional) types of metadata more important? Is technical metadata more significant than business metadata? Other factors to look at are the data modeling method used, level of integration supported, and the extent to which the metadata is exchangeable. This knowledge would assist organizations in identifying the key set of metadata modules (can be a mix of both technical and business metadata) for the core of the metadata repository when implementing homegrown solutions (see Homegrown Metadata Repositories in Section IV).

2. Can metadata improve the quality of data within and the performance of a data warehouse?

> Methods for measuring data quality dimensions such as accuracy, timeliness, completeness, and consistency, with specific reference to the data warehouses have been proposed [Hufford, 1996, Ballou and Tayi, 1999]. It is unclear though to what extent metadata contributes to good data quality and performance of a data warehouse. While investing in metadata is prescribed as a key factor for the success of data warehouses by several studies [Marco, 1998], [Inmon, 2000], [Sachdeva, 1998], none of these studies offer theoretical quantification or empirical support for measuring such impacts. Measuring this impact is a challenging task for several reasons. First, there are many alternative approaches for measuring the performance or success of a data warehouse. Some are based on the ease of managing a data warehouse focusing on technical administration [Hufford, 1996] and others are based on evaluating the end-user experiences [Wixom and Watson, 2001]. Second, there's no straightforward method for attributing costs directly to the metadata. As pointed out in the COTS product review (Appendix 1), metadata management components are embedded within other offerings and in many cases are not priced separately. It is also practically impossible to precisely assess the software development time allocated to metadata, since it is typically part of application programming efforts. Developing measurements methods for such study is likely to require fairly sophisticated models and methods for cost and benefit attribution [Stephens, 2004].

Metadata and Decision Making

Recent studies show that metadata does impact decision outcomes significantly [Chengalur-Smith et al. 1999], [Fisher et al. 2003]. Research issues linking metadata and decision-making are:

1. What elements of metadata affect decision-making?

> The differences in metadata functionality suggest that certain metadata elements may by more significant within certain decision scenarios than others. One can hypothesize, for example, that for certain types of decision tasks that require systematic processing and are data intensive providing the decision makers with quality metadata along with process metadata will make the decision-making process more efficient and improve its outcome.

2. What decision scenarios are likely to benefit from metadata?

Metadata is likely to be useful in a rational, data-driven, analytical decision-making scenario. We need to understand metadata's role within such scenarios. The use of metadata in other types of decision processes (intuitive, judgmental or bargaining) against rational and analytical decision processes should be investigated.

3. How do the decision roles affect the use of metadata and the benefit from it?

IT practitioners emphasize the need to manage the metadata repository to fit the day-to-day needs of the users [Marco, 1998, Inmon, 2000]. One can hypothesize that the use of the metadata dependent upon the decision-role. For instance, would operational/tactical decision-makers benefit more from the provision of metadata than strategic decision-makers? Empirical studies of data usage patterns indicate that usage style is influenced not just by the role, but also by other characteristics such as motivation, involvement, education level and professional experience [Chengalur-Smith et al. 1999, Fisher et al. 2003].. Studying the effect of the role and other characteristics on the use of metadata is important for understanding how to better manage metadata to fit the needs of the users.

4. What stages of the decision process benefit from metadata?

Modern decision-making models view decisions as a multi-stage process where preliminary cycles of elaboration and search precede the final decision. Metadata may influence not only the final outcome, but also the preliminary stages of data exploration and requirement definition. Understanding this process would help identify when (at what stage) and how much of metadata must be provided to decision-makers during the decision process.

Metadata and Knowledge Management

Metadata management is conceptually similar to knowledge management (KM). Creating metadata can be viewed as codifying data and creating a higher-level layer of knowledge for it. Similarly, codifying organizational knowledge within KM systems can be viewed as abstracting it into a metadata layer.

Nonaka [1994] suggests a framework of four knowledge creation modes:

• socialization	converting tacit knowledge to another form of tacit knowledge through interaction
• combination	converting explicit knowledge to another form of explicit knowledge,
• internalization	making explicit knowledge tacit
• externalization	making tacit knowledge explicit

Externalization is the heart of most knowledge management systems as knowledge in computer-based information systems should be codified and made explicit. Computerized systems for supporting knowledge management are based on the notion that tacit knowledge can be abstracted and documented and offer methods to turn tacit elements into reusable codified knowledge [Hansen et al.1999]. Nonaka's [1994] externalization process that makes tacit knowledge explicit can be viewed as creating a semantic abstraction, or creating a layer of metadata for it.

Markus [2001] looks at the following three purposes for creating knowledge as an element that plays a role in how knowledge is stored, processed, and distributed:

1. Self - knowledge for self-use, where little or no attention is paid to interpretable formatting.

2. Similar others - knowledge for others with a similar skill set. Assuming the ability of other users to assimilate knowledge easily, knowledge reuse efforts focus on providing essential details, rather than shape and format.

3. Dissimilar others - knowledge aimed for others without similar skill sets.

Assuming limited or no capability of target users to interpret the knowledge in its raw form, more efforts will be made to reconstruct and formalize it.

These incentive types and their implications on knowledge interpretability and formatting resemble our earlier discussion of metadata standardization and exchange. When metadata is aimed towards internal use by an application (self), no exchange is assumed and there is no attempt to standardize it. In case of metadata exchange among members of a product suite offered by the same vendor (similar others) a higher level of standardization across products is expected. Metadata exchange among products of very different nature (dissimilar others) needs a superior standard for exchange.

Sheth [2003] proposes the use of metadata for capturing knowledge. His study offers a methodology for creating layers of metadata to capture not only the basic business data entities, but also to structure them into business knowledge in the form of ontology – a shared conceptualization of the world as seen by the enterprise. The ontology consists of high-level business schemas, interrelationships between entities, domain vocabulary and factual knowledge. Knowledge is stored using a structured document and not a relational model.

The above discussion indicates that metadata management and KM have a lot in common. Bridging the two research areas could benefit both research streams. Some research questions to investigate are

1. Do similar theoretical foundations apply to metadata and knowledge management?

> Reviewing the knowledge management literature suggests that theoretical models developed for knowledge management may be relevant in the metadata context and vice versa. The conceptual similarity between the two areas is noticeable: metadata captures an abstraction of data, while knowledge management attempts to abstract organizational knowledge. Theoretical models for knowledge management may be applicable for metadata management.

2. What elements of metadata contribute to organizational knowledge management?

> Many metadata elements are valuable within a broader context of knowledge management. Process metadata, for example, offers further insight on the dataflow through business units and processes. A semantic data dictionary, another example, can be used as a tool for standardizing business terms. Possible reuse of elements between such systems is another topic to explore.

3. Should data warehouses, metadata repositories, and knowledge repositories be built on the same architectural foundations?

> Many knowledge management applications are built with a business unit or department focus and are thus limited in scale and scope [Hansen et. al, 1999]. Knowledge within each KMS must be interpreted using the thought models and vocabulary of the knowledge consumers for whom that KMS was built. This requirement makes it difficult to share and used the knowledge repository within a KMS across systems and departmental boundaries. Therefore, the architecture of knowledge management systems should bed rethought. This need exists within metadata implementations as well. Architectural solutions that address integration issues for KMS can be adopted for metadata repositories and vice versa. This will help standardize repositories of metadata and knowledge and permit better exchange and reuse of knowledge/metadata.

The questions about metadata are many and introduce a range of research opportunities. Much can be learned from using case studies describing implementation success/failures. Cases help identify practices that in turn can affect the success of future implementations. Addressing the above research issues is needed to understand the business value of metadata. Our hope is by drawing attention to much needed research issues on metadata we can motivate researchers to examine metadata more closely and more in-depth.

Editor's Note: This paper was fully peer reviewed. The article was received on April 8, 2004. It was with the authors for five weeks for two revisions. It was published on August ___, 2004

REFERENCES

EDITOR'S NOTE: The following reference list contains the address of World Wide Web pages. Readers who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that

1. these links existed as of the date of publication but are not guaranteed to be working thereafter.

2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.

3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.

4. the author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Ballou, D.P and Tayi, G.K. (1999), "Enhancing Data Quality in Data Warehouse Environments", *Communications of the ACM,* (42) 1, January

Blumstein G. (2003), "Metadata Management Architecture", *DM Review*,(13)8 August

Cabibbo L., Torlone R. (2001), "An Architecture for Data Warehousing Supporting Data Independence and Interoperability", *International Journal of Cooperative Information Systems*, (10) 3 pp. 377-397

Chengalur-Smith I., Ballou, D. P., and Pazer, H. L. (1999), "The Impact of Data Quality Information on Decision Making: An Exploratory Study", *IEEE Transactions on Knowledge and Data Engineering*, (11)6, November-December, pp 853-864

Fisher C. W., Chengalur-Smith I., and Ballou D. P. (2003), "The Impact of Experience and Time on the Use of Data Quality Information in Decision Making", *Information Systems Research*, (14)2, June pp-170-188

Hansen, M.T., Nohria, N. and Tierney T. (1999), "What's Your Strategy for Managing Knowledge?", Harvard Business Review (77)2, March-April

Holland, P. (2000), "Virtues of a Virtual Data Warehouse", *Datamation*, 27(3) March; http://www.itmanagement.earthweb.com/datbus/article.php/621401

Hufford, D. (1995), "Data Warehouse Quality", *DM Review*, (6)1 **,** January.

Imhoff, C., Galemmo, N. and Geiger, J. G. (2003a), *Mastering Data Warehouse Design: Relational and Dimensional Techniques*, New York: Wiley

Imhoff, C. (2003b), "Take a Trip and Never Leave the Farm: Virtual Data Marts in CIF", DM Review (13)1, January

Inmon, B. (2000), "Enterprise Metadata", DM Review(10)7, July

Jarke, M., Jeusfeld, M., Quix, C., and Vassiliadis, P. (1999) “Architecture and Quality in Data Warehouses: An Extended Repository Approach”, *Information Systems*, (24) 3, pp. 229-253

Jarke M., at al. (2000), *Fundamentals of Data Warehouses,* Berlin: Springer-Verlag

Kimball, R. (1998), "Meta Meta Data Data", DBMS, (11)3 March .

Marco, D. (1998), "Managing Meta Data", DM Review (8)3 March.

Marco, D. (2000), *Building and Managing the Meta Data Repository: A Full Lifecycle Guide*, New York: Wiley

Markus, M. L. (2001), "Towards a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success", *Journal of Management Information Systems*, 2001, (18)1, Summer,pp 57-93

Mimno, P. (2002), “How to Avoid Data Mart Chaos Using Hybrid Methodology”, TDWI FlashPoint, June 20, http://www.mimno.com/articles/article5.htim

Moncla B. (1999), "Business Metadata Integration", DM Review (9)9. September

Nonaka I. (1994), “A Dynamic Theory of Organizational Knowledge Creation”, Organization Science (5)1, February

Sachdeva, S. (1998), "Metadata Architecture for Data Warehousing", DM Review (8)4,April

Seeley R., and Vaughan J. (1999), "Meta is the word", *Application Development Trends*, http://www.adtmag.com August

Shankaranarayanan, G. and Watts-Sussman, S. (2003), "A Relevant Believable Approach for Data Quality Assessment", Proceedings of the MIT International Conference on Information Quality (IQ 2003), Boston, MA. October

Shankaranarayanan, G., Ziad M., and Wang R. Y. (2003), "Managing Data Quality in Dynamic Decision Making Environments: An Information Product Approach", *Journal of Database Management,* 14(4), Oct-Dec pp 14-32.

Sheth, A. (2003), "Semantic Metadata for Enterprise Information Integration", *DM Review* (13)7, July

Stephens, T. R. (2004), “Knowledge: The Essence of Metadata: The Meta Data Experience”, *DM Review* (14)3, March

Vaduva, A., and Vetterli T. (2001), "Metadata Management for Data Warehousing: An Overview", *International Journal of Cooperative Information Systems*, (10)3 pp. 273-298

Wang, R. Y., Lee, Y. W., Pipino, L. L., and Strong, D. M. (1998), “Manage Your Information as a Product”, *Sloan Management Review*, 39(4), Summer

White C. (1999), "Managing Distributed Data Warehouse Metadata", *DM Review,* (9)2 February

Wixom, B.H. and Watson, H.J. (2001), “An Empirical Investigation of the Factors Affecting Data Warehousing Success”, *MIS Quarterly,* (25)1, March, pp. 17-41

APPENDIX I. SOFTWARE VENDOR OFFERINGS

Note: This appendix discusses offered by software vendors that apply to metadata in the data warehouse. The data were obtained in February 2004. Successor products by the same vendors or by new vendors may apply when this Appendix is consulted. The limitations on URL’s given at the beginning of the Reference section also apply to this appendix.

Data Storage Products

Relational database management systems (RDBMS) are the leading data storage technology for data warehousing. Oracle (http://otn.oracle.com/), the dominant software vendor in this market, provides storage and other warehouse functions such as ETL and business intelligence. Version 9i of Oracle provides support for metadata that is captured in what Oracle calls the Warehouse Builder Repository.

The metadata is stored in a set of relations and is primarily data dictionary metadata (see Metadata Functionality(Section II)). The relations map warehouse data elements to underlying operational databases and their schemas. The Warehouse Builder Repository supports creating metadata definitions for existing data sources. It also supports importing database design information from data modeling tools such as the Oracle Designer, Oracle's CASE (Computer Aided Software Engineering) tool, and ERWin, a data modeling product from Computer Associates that supports multiple database platforms.

The metadata repository is integrated with other Oracle data warehousing products: ETL, OLAP server and reporting tools. The repository includes a set of built-in reports on metadata contents. It allows metadata exchange with external products using the Common Warehouse Model[9] (CWM) Microsoft MS-SQL Server (http://msdn.microsoft.com/) is gaining popularity for small to mid-range data warehouse implementations. MS-SQL Server (2000 and newer versions) includes Metadata Services, a component that supports metadata management. Like Oracle, SQL Server's metadata is stored in a relational structure and is data dictionary metadata that is geared towards integration with other Microsoft products. SQL Server supports integration with other RDBMS products, and adopts the Open Integration Model[10] (OIM) for metadata exchange. It also supports XML (eXtensible Markup Language) and COM (Microsoft's Common Object Model) exchange formats.

The Sybase (http://sybooks.sybase.com/) server, primarily UNIX based, is a strong contender in the financial sector. While Sybase ASE, the traditional Sybase database product, is geared more toward transactional systems, the product for data warehousing is Sybase IQ, which employs unique storage (relational tables ordered by columns rather than by records) and indexing techniques for high-speed querying and aggregation of large datasets. Sybase products follow the Adaptive Server Anywhere schema – a conceptual schema for implementing relational databases. Within this schema, data dictionary metadata is captured in a set of system tables. This set is referred to as the "Catalog Store" in Sybase IQ. Newer versions of Sybase support the Common Warehouse Model. The Sybase tool for metadata integration is the Warehouse Studio, an advanced modeling suite that can communicate with the metadata layer of many other data sources through the Meta Integration Model Bridge (MIMB) utility.

UDB (formerly DB2), a database management product offered by IBM (http://publib.boulder.ibm.com/infocenter/), manages metadata in its Information Catalog Center. UDB supports data dictionary metadata besides capturing business definitions of the data, such as description of data elemens in business language that can be integrated with front-end applications. The Information Catalog also uses relational structure for storage and supports the Common Warehouse Model for metadata exchange.

Teradata (http://www.teradata.com), a division of NCR, offers the Teradata database server, which is popular primarily for very large-scale (terabyte and above) data warehouses. Similarly to the other RDBMS vendors, Teradata takes the repository approach in implementing the metadata layer. "Teradata Meta Data Services", the metadata layer offered by Teradata, is stored in a centralized set of database tables. It supports the CWM standard and can be accessed via

[9] Described under "Metadata Management Using Data Warehousing Tools" (Section IV)
[10] Described under "Metadata Management Using Data Warehousing Tools" (Section IV)

multiple methods – ODBC, API or XML. The metadata layer also offers utilities for graphical presentation and reporting, that can be useful for business users.

OLAP Servers for Analysis

OLAP (On-Line Analytical Processing) servers offer an alternative approach for storage. Data storage is optimized for data warehousing use: fast querying and aggregation over large data sets. OLAP servers permit the creation of a multidimensional "cube" of attributes (dimensions) and measures (facts) that can be interfaced with reporting tools for high-efficiency reporting. OLAP products are offered by all the leading RDBMS software vendors: Oracle\OLAP, MS-SQL OLAP Services, Sybase IQ, and UDB OLAP Server. A feature common to all these OLAP products is that the metadata layer is identical to the one used by their corresponding RDBMS products, including storage method (relational structure) and exchange interfaces.

Hyperion (http://dev.hyperion.com/) offers Essbase, a non-relational OLAP server (although it offers an SQL interface). Unlike OLAP offered by RDBMS vendors, the metadata for Hyperion is not stored in a relational structure, but in a proprietary one. The metadata can only be accessed via its own Application Programming Interface (API). Essbase supports the Common Warehouse Model or metadata exchange but to a limited extent. Hyperion concentrates only on dimensional elements of an OLAP cube.

Other reporting tools such as Business Objects, Cognos, and MicroStrategy use proprietary OLAP cubes, which are internal in their software and are tailored to support their front-end reporting utilities. These cubes cannot serve as independent data sources. Reporting tools are discussed in more detail later in this Appendix.

An emerging hybrid is the object-relational databases, gaining popularity with the growing interest in the web services. The basis of this technology is extending relational databases to include object structures that can be used more efficiently for object-oriented programming. A tool that is typically used for mapping between the object-oriented elements and the relational structure is XML. An XML-structured file includes both data and metadata, thus allowing efficient communication regardless of the data model used by databases at either end. All the leading RDBMS vendors (IBM, Oracle, Sybase and Microsoft) offer bi-directional utilities both for parsing XML data files into relational tables and for extracting XML files out of the relational structure. The metadata layer used for XML conversions is the same one as that used by their corresponding database products and deals with data dictionary metadata. The protocol for metadata exchange via XML is XMI (XML Metadata Interchange). Other vendors offer products that were built as object-relational rather than extending existing RDBMS. First SQL and MDBS\Titanium are two examples. The information in product web pages and white papers is insufficient to determine the data structure used by these products to capture metadata or to determine the standards (CWM or OIM) used by these products for metadata exchange.

ETL (Extraction, Transformation and Loading) Products

All major suppliers of tools for automated ETL (Extraction, Transformation, Loading) support metadata. Unlike database products, where the metadata focused on the data dictionary elements, ETL tools must address process metadata as well as mapping between source and target data elements and job scheduling. Automated ETL tools are offered by Oracle, Microsoft, and IBM as well as by vendors who specialize in ETL tools. Database vendors typically offer ETL utilities. Oracle offers ETL functionality within the Warehouse Builder, Microsoft offers the DTS (Data Transformation Services) suite tied to SQL Server, and IBM offers ETL tools in Warehouse Manager of UDB. In all cases, the metadata used for ETL is tightly integrated with the metadata repository used by their corresponding database product. Vendors offering business intelligence products such as Business Objects and Cognos also offer tools for ETL. Cognos, for example, offers DecisionStream, an ETL software tool that is configured to work with other Cognos products. DecisionStream uses a proprietary metadata structure that is internal to the product and metadata stored within cannot be integrated or exchanged with metadata captured using other vendor-products.

Informatica's (http://www.informatica.com/) ETL tool claims to be product independent. Informatica is capable of communicating with all the leading database products both as sources and as targets. It can also interface with data in flat-file structures and potentially with all other types of data sources via programmable interfaces. The metadata layer in Informatica is stored in a relational structure and consists of process metadata and to a lesser extent data dictionary metadata. Informatica supports the Common Warehouse Model for metadata exchange with data storage products (any ODBC-supported relational source and also Essbase OLAP server) and with modeling tools (ERWin, Sybase Power Designer, Oracle Designer).

Hummingbird's (http://www.hummingbird.com) ETL tool is product independent. It is a "lighter weight" than Informatica and geared towards small data warehouse implementations. Similarly to Informatica and other ETL tools, Hummingbird stores its metadata layer, called the "Met@Data", using a relational structure. The metadata serves the internal needs of the product and is not easily exchangeable.

Business Intelligence Products

Front-end utilities for reporting and data analysis are significant DW components from the business end-user viewpoint and are commonly claimed to be Business Intelligence (BI) tools by the commercial vendors. While the older generation of reporting tools (such as Crystal, Focus and PowerBuilder) interfaces directly with the RDBMS through SQL queries, the new generation typically uses a semantic metadata layer. Elements of the data dictionary metadata, such as tables and fields, are exposed to the end-user, not directly but via a graphical representation that also provides the business context. Metadata in those tools must address not only the storage aspects of the data but also the presentation aspects – business interpretation and vocabulary, formats, layouts, templates, and aggregation preferences.

Such reporting tools are offered by data storage vendors such as Oracle, IBM, and Microsoft and by ETL vendors such as Informatica. These vendors use the metadata repository that is part of the corresponding database or ETL tool. Other companies offer product independent reporting tools. Such tools require a more sophisticated metadata layer, to support both the RDBMS products as well as other data sources. A common feature in these tools is the concept of "Internal Cube" – the data is restructured into a multi-dimensional cube, optimized for reporting and on-line analysis. The tools may differ in terms of where the cube is stored – some require a centralized, server-based, storage (MicroStrategy) while others allow cube creation on the client (end-user) machine (Business Objects, Cognos). Besides the data dictionary metadata, the more advanced reporting tools offer metadata support for administration such as user setup, security, authorizations, report distribution and usage tracking.

MicroStrategy (http://www.microstrategy.com/) is a ROLAP (Relational OLAP) tool, geared towards data stored in RDBMS data sources. The tool uses a centralized metadata architecture, where all metadata components are installed on a single central server and stored in a relational structure. The metadata includes many semantic layers – from basic mapping of the databases elements to warehouse elements and warehouse to presentation elements, through definition of dimensions and measures, to more sophisticated data consolidation and filtering techniques that may result in multiple SQL query passes. While MicroStrategy does use the metadata available in the RDBMS sources, it doesn't support the notion of metadata exchange. MicroStrategy does not support the Common Warehouse Model or the Open Integration Model. Some limited metadata integration with Informatica is supported using a proprietary data exchange facility.

Business Objects (http://www.businessobjects.com/) supports a broad range of data sources – not only RDBMS, but also OLAP servers, and other data sources such as flat-files via APIs (Application Programming Interface). It offers a flexible reporting structure that can use a centralized metadata repository on the server and/or custom repositories in the client. As a result, its metadata architecture comes in two interchangeable flavors – centralized server-based metadata repository stored in a relational structure and the desktop-based repository, stored in a local (non-relational and proprietary) file. Like MicroStrategy the metadata covers both the mapping of database elements to the warehouse elements and mapping the warehouse data elements to the presentation elements. Unlike MicroStrategy, Business Objects supports metadata exchange using the Common Warehouse Model.

Cognos (http://www.cognos.com/) is another business intelligence tool vendor that offers both server-based and local metadata repository architectures. The metadata used by the reporting utilities is the same one used by Cognos' DecisionStream – its proprietary ETL tool. The information provided in the commercial website points out that the metadata integration within Cognos's products is robust, but it is not clear to what extent the product supports standards for metadata exchange with other external products

Metadata Management Tools

Since 2000, a growing number of products are being marketed that specialize in metadata management. These products claim to be vendor and technology independent. Such tools are aimed to bridge some of the major gaps with using metadata capabilities of commercial data warehousing products. They provide a wide range of metadata functionality, with emphasis on business metadata. They attempt to support multiple exchange standards and support integration with all leading data warehousing products. A drawback of using such tools is the investment required for software licensing and for implementation. Data warehousing products typically provide their metadata management utilities at no additional cost. Even with the presence of a metadata management tool, a firm still needs to establish and maintain a metadata layer within its data warehousing tools and integrate this layer with the repository in the metadata management tool. Despite these concerns, we believe that being well supported by most other data warehousing tools, such products can be an attractive choice for metadata management.

MetaCenter by Data Advantage Group (http://www.dag.com/) supports the capture of different types of metadata including contextual metadata that can be extracted from a broad range of information systems and technologies- transactional, networking, decision support, data management, and data quality. It even supports the capture of metadata from data modeling tools, software development tools, operating platforms and legacy applications. The suite includes tools offering a wide range of functionalities: data dictionary, thesaurus, naming conventions, business rules, requirements, personnel skills and responsibilities, IT infrastructure, and security. It allows integration between all those metadata elements, as well as integration with metadata provided by external products (e.g., IBM, Business Objects, Oracle, Microsoft, and Informatica) A set of metadata exchange models (Common Warehouse Model, XML Metadata Interchange, and Meta Object Facility) is supported for both importing and exporting metadata. On top of the integrated metadata layer, which is stored in a relational database, MetaCenter also provides a set of GUIs for defining reports off of the metadata as well as analysis tools for extracting information from metadata.

MetaBase by MetaMatrix (http://www.metamatrix.com) provides capabilities similar to MetaCenter, including support for metadata exchange protocols (CWM, XMI, MOF, and UML) and data analysis capabilities. However, MetaBase emphasizes technical metadata, specifically those needed for data integration.

Ascential (http://www.ascential.com) takes an integrated metadata approach with MetaStage, its metadata management product. MetaStage consists of the MetaHub directory, a centralized repository that stores the metadata. Built on top of it is the Explorer, a suite of GUI supported tools that allows administrators, as well as end-users, to navigate, analyze, publish, and subscribe to metadata components stored in MetaHub. MetaBrokers are metadata exchange utilities that provide "hooks" for connecting with other data warehousing products and allow bi-directional metadata transfer between the MetaHub and other tools. MetaStage supports most common metadata exchange formats and can communicate with tools provided by all the leading vendors. MetaStage supports both technical and business metadata and uses metadata for data-process mapping and automated quality improvement.

URLS FOR PRODUCTS

Ascential, http://www.ascential.com/

Business Objects, http://www.businessobjects.com/

Cognos, http://www.cognos.com/

Data Advantage Group, http://www.dag.com/

Dublin Core Project, http://www.dublincore.org/

Hummingbird, http://www.hummingbird.com/

Hyperion, http://dev.hyperion.com/

IBM, http://publib.boulder.ibm.com/infocenter/

Informatica, http://www.informatica.com

MetaMatrix, http://www.metamatrix.com

Microsoft, http://msdn.microsoft.com/

MicroStrategy, http://www.microstrategy.com/

NSDL, http://www.nsdl.org/

Oracle, http://otn.oracle.com/

Sybase, http://sybooks.sybase.com/

Teradata, http://www.teradata.com/

LIST OF ACRONYMS

API	Application Programming Interface
BI	Business Intelligence
COTS	Commercial Off-The-Shelf
CSS	Cascade Style Sheets
CWM	Common Warehouse Model
DSS	Decision Support System
DW	Data Warehouse
ETL	Extraction, Transformation, and Loading
GUI	Graphical User Interface
IP	Information Product
IPMAP	Information Product Map
KM	Knowledge Management
KMS	Knowledge Management Systems
MDC	Meta Data Coalition
MOF	Meta Object Facility
ODS	Operational Data Store
OIM	Open Integration Model
OLAP	On Line Analytical Processing
OMG	Object Management Group
RDBMS	Relational Database Management System
ROLAP	Relational OLAP
UDB	Universal Database (IBM Product)
UML	Unified Modeling Language

VDW	Virtual Data Warehouse
XML	eXtensible Markup Language
XMI	XML Metadata Interchange

ABOUT THE AUTHORS

G. Shankaranarayanan[11] obtained his Ph.D. in Management Information Systems from The University of Arizona in 1998. He is assistant professor of Information Systems in the Boston University School of Management. His research interests include schema evolution in databases, data modeling requirements and methods, and structures for the management of metadata. Specific topics in metadata include metadata implications for data warehouses, metadata management for knowledge management systems/architectures, metadata management for data quality, metadata models for mobile data services and for managing security for mobile data access. He is responsible for the Mobile Consumer Laboratory at Boston University School of Management.

Adir Even obtained his M.B.A. in Business Administration from Tel-Aviv University School of Management and his M.Sc. in Computer and Electrical Engineering at the Technion of Israel. He is a doctoral student at the Boston University School of Management. His research interests include business valuation of information systems and metadata, studying implications of system architectures, data quality, data warehousing, and knowledge management. Prior to pursuing doctoral studies, he worked as a programmer and as a senior software development manager designing and implementing data warehouses and decision support solutions.

[11] *Editor's note*: He is known as Shankar, for short.

Communications of the Association for Information Systems

ISSN: 1529-3181