INFORMATION-CENTRIC VS. STORAGE/DATA-CENTRIC SYSTEMS

Charles Milligan
Sun Microsystems, One StorageTek Dr, Louisville, Colorado, 80028, USA

Steve Halladay, Deren Hansen
Cogito, Inc., 170 West Election Drive, Suite 205, Draper, UT 84020, USA

Keywords: Information Management, Graph Database, Information Modelling, Interpretive Frameworks.

Abstract: It is essential to recognise that information (i.e., the meaning and value that must be extracted from data for a business to run) is very different from the data itself. Information must be managed using different processes and tools than those used in data management. The current notion of Information Lifecycle Management (ILM) is really only about making Systems Managed Storage work universally and does not relate to information management at all. However, recent developments of new technologies have potential to open a new paradigm in extracting, organizing and managing the meaning and value from data sources that can allow processes and decision systems to take a quantum leap in effectiveness. The networked structure of a graph database combined with concept modelling will foster this shift.

1 INTRODUCTION

The word 'information' has been used almost interchangeably with the word 'data' throughout the storage industry. The terminology "ILM" or "Information Lifecycle Management" is also used throughout the storage industry to describe the latest strategy for managing storage resources. However the tools, products, and processes used to invoke ILM strategy are not about information. For the purposes of this paper, the term information will be used exclusively to identify the valuable concepts and meaning buried in data that must be extracted organized and processed in order to run a business or make decisions. We will always differentiate between the word 'information' and the word 'data'.

This paper proposes a system for managing the abstract entities that embody information independent from the management of the concrete objects that comprise the data in which the information is hiding (e.g., files, databases, documents, email, etc.). A graph database system is described which along with a concept modeling process (Milligan, 2005) holds promise to change how business processes interact with data, making the processes more effective by orders of magnitude.

2 INFORMATION VERSUS DATA

2.1 Value Systems are Different

The value of data is generally measured by the owner of the data in three dimensions. The first is the cost to the business of losing the data. The second is the expense incurred in accessing or using the data. The third is the infrastructure and administration costs associated with keeping the data lying around for long periods of time. There are a number of very specific requirements associated with each of these that have in some instances become onerous to bear. There are standards bodies for process control like ISO9000 and government legislation like Sarbanes-Oxley that document specific processes for managing data and making it available for inspection. These in turn drive significant data management costs and administrative costs that are a burden on profitability. ILM is all about making these processes work significantly more efficiently, but

Milligan C., Halladay S. and Hansen D. (2006).
INFORMATION-CENTRIC VS. STORAGE/DATA-CENTRIC SYSTEMS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 501-507
DOI: 10.5220/0002493305010507

have nothing to do with information management as is discussed in section 3.

The value of information on the other hand lies in the ability to perceive, understand or identify the most effective way to proceed (i.e., make decisions or invoke processes), especially when confronted with a dilemma. In the absence of a dilemma, most organizations are complacent and are blindly content to continue their standard data processing extraction against their captive data. They are satisfied with the information available thereby. However, the internet accessibility of vast quantities of information rich data has changed the relative value of captive data sources and consequently introduced new processes, some of which have become new verbs in our vocabulary (we now talk blithely about googling something as a metaphor for doing an internet search). However there are three problems with the current technology associated with such searches. Generally we are drowning in data and fighting:

Precision – how to sort out all the extraneous information included in the results from that which is specifically relevant so that integration of results is meaningful,

Recall – how to identify that we have found and can access to all of the information that is truly relevant to a particular process or problem,

Integration – how to discover the existence of composite information that only exists by combining apparently redundant or irrelevant input from a number of seemingly independent information nodes.

These problems cannot be overcome by the ILM techniques described below. They must be attacked in an entirely new way that allows the threading of relevance to occur across broad reaches of data storage including the vastness of the internet and emerging Grid infrastructures.

2.2 Information Identification

Values (i.e., specific decoding of instances of data) in isolation - like the number ‘150’ or the letter ‘B’ – are not very helpful. However, when we put the values into a context, such as “IQ = 150 but skill level = B”, or “bank account balance = 150B”, then we can derive some usefulness from the values. It is precisely the context of data decode that makes the result useful or, in other words, gives the data its meaning.

Put more precisely, context provides an interpretive framework and information arises from the interpretation of data. Indeed, without an interpretation, data has no meaning. An extreme example of this is with encrypted data. Without the context of the particular encryption algorithm and the metadata called the ‘key’, the data is left with no intrinsic meaning. Boldly stated, encrypted data by itself has no information content whatsoever. Often the interpretation includes both explicit and implicit elements (e.g., with data encryption, the algorithm may be implied by the size of the key while the specific key metadata is required input for the decryption). Moreover, the implicit elements often cross arbitrary data boundaries. Consider the examples given above: The context of IQ might lead to an interpretation of independent data decode in this context to indicate an intelligent person ultimately capable of a great deal while the companion concept of skill level would imply that same person is not yet trained, on the other hand the context of bank account combines the two data decodes and might lead to an interpretation of the availability of exorbitant funds. Both contexts invoke larger networks of conceptual associations (e.g., IQ as a measure of intelligence and skill level is a measure of training or experience and both intelligence and experience are desirable) that form the interpretation. Different conceptual networks will lead to different interpretations of the same data.

Interpretation is a function of the relationships that impinge upon the object in question. Data only provides evidence of the existence of a thing. Information arises when we understand that thing in relation to other things. For example, data indicating the existence of a hungry tiger may be of some esoteric interest, but knowing that said tiger has the relationship of “behind” the object “my door” provides important information. Relationships, then, are the atomic units of information.

2.3 Information Management vs. Data Management

The management of data is an exercise in managing discrete objects and sets of objects. These are concrete in nature because every bit of data takes up some physical space on a piece of media (in a cartridge or a device). The objects can be moved individually or in groups and a great deal of metadata must be generated to keep track of them. The metadata is generally about the object (size, location…) or the environment in which the object has been created or used (creation date, expiration date, date last used…). There is now a movement in industry to create additional metadata about the contents of the data which are for the most part sophisticated versions of key word indexes. None of this embodies the information contained in the data although the content metadata is a first approximation. For the most part, the content metadata simply copies a subset of the information content of the data and then indexes where such information might exist. It does not present or appreciate the information itself. The fact that all information content can be extracted from the data (and the data must still be managed) as has been noted already for encrypted systems is the real eye opener to the differences between information and data.

Information management emphasizes the relationships between data objects. The notion of a set is a natural representation for the collection of discrete objects that are the subject of data management. However, networks are the natural representation for information management. Efforts in the industry along the lines of configuration management and product lifecycle management are a small step toward information management.

3 TOWARDS INFORMATION MANAGEMENT

3.1 ILM a Misnomer

Information Lifecycle Management (ILM) was coined as a description of a strategy by Storage Technology Corporation in 2002. The CIO's attending the 2002 Forum gave instant, powerful and positive feedback because it was clear that this was an efficient tiered storage management concept under an umbrella for all resources storing data at the best possible economy (an idea that has been around since 1979). Prior to 1979 the typical storage administrator in large enterprise systems like IBM sold could effectively manage about 11 gigabytes (about 32 disk drives) of storage with an allocation efficiency of about 35% (obvious inefficiency) (Chalfant, 2005). However, such an administrator was required to manage far more than that and it was growing (similar to today). IBM put together a systems strategy (not a product) called Systems Managed Storage or SMS and today that same administrator can manage 30 terabytes (about 200 disk drives) at 80% allocation (a clear improvement). The primary focus of ILM now is to address the same issues that SMS did 25 years ago, but for the open systems market where today's administrators can only manage 300 gigabytes (a handful of disk drives) at 40% allocation (Chalfant, 2005).

The basic ingredients of an ILM architecture and solution are well understood. The first requirement is a classification engine to define the value of data (mostly unstructured i.e., not in a database). Next is a migration engine that can manage the movement and compression of data. This is coupled to a high-speed data mover which actually moves data up and down a hierarchy (a hierarchy is presumed). In addition a system for providing data archive and protection must be in place. Also because of the need for providing increased levels of security and structure for all of the unstructured data we need a security engine that provides for legally compliant protection (such as Write Once Read Multiple or WORM that can also provide audit trails to monitor reference and usage by process or person) and also doubles as part of the classification engine by providing metadata that describes the unstructured data. There is also a need for protecting data at rest so encryption must also be available.

For all of these things to be efficient without vast amounts of human intervention, a policy engine that can automate placement decisions is required. Finally, a multi-tiered storage infrastructure allows the cost of storing data to match the business value of that data.

So it is clear that ILM is not about information but rather about efficiently managing a storage system (devices management and data management)

3.2 Barriers to Getting There

At one level, it is natural that that industry focused first on the basic issues of data representation and management. Without reliable means of creating, storing, and managing data, there is no reason to be concerned with information management. But the problems of physical data handling have been well understood for a decade or more. During that time, the lines of inquiry have not moved much beyond data representation and management because we have lacked the tools (conceptual and implemented) to move into information issues. A portion of that lack can be attributed to the fact that the table/column model, most broadly expressed in the form of relational databases, has been so successful that it has starved other approaches.

3.3 Enablers for Information Management Tools

In order to begin the transition to information-centric systems, we need to adopt new concepts and implement new forms of representation. At the conceptual level, we need to move beyond set-based approaches in which the members of a set must be homogenous and consider networks of connections that can capture the idiosyncratic relationships of particular objects of interest. The conceptual changes will, however, be no more than an academic exercise if we cannot build systems that, in time, can be as highly optimized for information management as the widely deployed, set-based data systems are for data management. A network based representation of information using a graph-based approach may be the way to get beyond the storage and data-centric systems. (Lee, 2001)

4 GRAPH REPRESENTATION EFFECTIVE FOR INFORMATION

4.1 Relationship Paradigm

When we connect data items with relevant relationships and thereby create a context for the data, we embody information. The essence of the context is the relationship of data values to labels. The associated label and context give the value meaning by giving us a conceptual pathway for navigation.

For many years computer scientists have recognized the usefulness of this specific type of relationship. They unabashedly assign values to labels and store the tuples for later reference. The relationships can be many to one or one to many making for "n"tuples. When computer scientists create a collection of n-tuples (i.e., the labels and values) they call it a database. (Codd, 1970).

Fundamentally databases, constructed of these n-tuples, have proven to be very useful. The computer industry has enhanced the usability of these label/value sets by sorting the sets and creating indices to help find specific instances.

While the indexing relationship used by databases causes a significant performance improvement, the logical relationships used by standard databases (i.e., relational operators, Boolean operators, etc.) are a limiting subset of the natural relationships required to be able to express many concepts. In order to increase the relationship richness, and thereby increase our expressive power, industry has added Object Oriented relationships to data modeling capabilities. These relationships include encapsulation, inheritance, aggregation, etc. Armed with these additional relationship types it is possible to successfully develop much more sophisticated and expressive systems.

While Object Oriented approaches enhance relationship-richness over the simple relationships in databases, Object Oriented approaches also fundamentally bound relationship richness. UML, a standard Object Oriented modeling language (UML reference), exhibits these limitations in that UML has no general relationship representation mechanism. Instead, UML has symbols for specific types of relationship. For example, UML class diagrams represent inheritance with a triangle, aggregation with a diamond outline and composition with a solid diamond. In addition, UML class diagrams represent attribute values as actually part of the entity. UML class diagrams allow representation of other ad hoc relationship types, but the ad hoc relationship types are not diagrammatically significant like those previously mentioned. The result is that UML class diagrams tend to focus attention on the canonized relationship types.

Semantic networks have rich, uninhibited relationship representation. (Sowa, 2002) Usually semantic networks consist of entities with free form relationships. No relationship types are considered elite; so all relationship types have similar representations. While semantic networks are extremely relationship rich, a common weakness in semantic networks is the confusion between the entity and the word that names the entity. Using the word as the entity limits the conceptual representation in much the same way as the class attributes in the UML class diagrams. In many cases, this confusion may not have a significant effect. However, this common mistake robs the conceptual representation of its language independence. Language, and its mapping to entities, is really just another type of relationship.

The medium aspect of the ideal representation mechanism consists of the building blocks of the conceptual representations. In some sense, the medium is similar to syntax for the representational mechanism. Fundamentally these building blocks are entities and relationships. Graphical network representations are intuitive presentations for these conceptual representations where nodes represent entities and arcs or edges represent the relationships. To facilitate working with the representations, entities and relationships have labels, however, the labels are merely for the purpose of discussion and do not suggest the “meaning” of the node or relationship. Nodes and edges derive their meaning strictly from their connections.

4.2 Graph-based Information Representation

Graphs, as a mathematical construct, have been studied for hundreds of years. More recently, graphs have been applied to practical problems involving networks, particularly in transportation and communication. The key observation is that network problems focus, not on the things, but on the nature of the connections between things. The essential information in the traveling salesman problem is not the destination cities, but the ways in which those cities are connected in a transportation network and the cost of making a trip between two particular cities. As we observed earlier, the interpretive frameworks that enable us to operate in terms of information emphasize relationships. A graph-based representation is the natural choice for expressing relationships. (Ebert, 1996)

In a graph-based information representation scheme nodes are labeled end-points that represent a single, atomic entity and arcs represent an assertion of an association between two nodes. Arcs are typed so that multiple associations may be expressed in a single representation. Values may be associated with each node to carry information that may be needed at other levels of the system (e.g., a string label to be displayed to a user) but are treated, insofar as the graph representation is concerned, as opaque blocks of data.

Because information is expressed in relationships, systems that implement a graph-based information representation will be optimized to store and manage networks of relationships. Graph theory considers directed and undirected arcs. We have found that a pair of directed arcs, where one arc points from the first node to the second and another points from the second to the first, gives us a general construct that can be used as either a directed or undirected connection. More importantly, this representation captures the fact that if we can assert that one object has a relationship with another, we also implicitly assert that the other object has a reciprocal relationship with the first. By making the reciprocal relationship explicit the graph-based representation naturally provides back-links that double the possible traversal patterns. We call this construct a relationship.

With the majority of the information residing in the networks of relationships, nodes must represent single, finer-grained entities. Because any two nodes in a graph may be linked by a relationship, a concept need only be expressed once and represented by a single node. This has the important side-effect of naturally creating a fully-normalized representation.

The notion that nodes represent atomic entities can be a difficult concept. In a graph-based information representation scheme, each node should represent one, atomic thing. In practice, this generally means that what would be an object in an object-oriented system or a row in a relational system would be a network in a graph: The graph representation of, say, an employee record would have a node for each field in the record and all of those nodes would be connected with the node that represents the employee record.

4.3 Performance of Graph-Based Information Representation

Extracting precise information can be time consuming and expensive when working with complex data sets. For example, consider the following comparison of using a modeling paradigm implemented in a graph based storage system versus the current solutions in a relational database system. Administrators using relational database technology strive to optimize queries across multiple tables, but this often involves iterative cycles for filtering out irrelevant information and structuring statements that reduce the answer set based on ordered sequences. Because of this, relational queries through chained data are often limited to four or five connection levels. In many cases, a four or five degree search becomes unmanageable, overly time consuming, and requires additional hardware and software.

Queries when using a graph database are significantly simpler, with the ability to traverse data that was never de-structured to fit into tables. To a large degree, data in a graph follows its natural pattern of existence with relevant information related through close association. This pattern follows even as the data is committed to disk.

To illustrate, assume a large data set with records indicating parent-child relationships but no extended family relationships. The objective is to find a common ancestor among two individuals who are not known to be related. To search parents on both sides going back seventy-two generations requires a search with 2^72 iterations. Given standard server class hardware and relational database technology, the problem can take up to three hours. The same exercise with graph technology can be performed in under a second—the operation is a simple node walk to find a common ancestor. A graph database makes parent/child relationships inherent in the data structure and closely located through arcs. This same query performance is possible with any type of related information. The result is an execution time that is 4 orders of magnitude shorter on average. (Clegg, 2005)

5 INFORMATION MANAGEMENT VIA MODELING

5.1 Models

A graph-based information representation is best understood as a model. A model, according to the dictionary is, "a form in miniature, in natural size, or enlarged of something to be made in similar proportions." Implicit in the act of creating a model is a need for a tractable thing that represents some or all of another thing. The process of selecting some aspects for emphasis while ignoring others is an essential part of modeling. Take, for example, road maps which generally show the drivable network but do not show the buildings and features along the roads. In a similar fashion, the information represented in a graph is a subset of all possible information that could be represented. The choice of what to represent and what to ignore is an act of interpretation.

The choice of representation is often informed by the process of abstraction. An interpretation of some data begins with organization. In the course of organizing the data, one begins to note entities that can be grouped together because they share some commonality. The burden of the model, then, is to represent those groups as abstract entities and to capture the dynamics expressed in the data in the form of relationships between those abstract entities. Abstract models are often used for simulation.

The issue of what to represent also depends on the level of detail of the model. Going back to the example of the road map, with the advent of internet tools to formulate significant quantities of data into maps we now have the level of detail as an orthogonal vector stipulated by the user at run time. So the act of interpreting becomes a sub-process to the higher level act of specifying the interpretation domain which links to a higher level process that is the human problem initially being addressed.

5.2 Meta Models

The interpretation of some body of data can be formalized and made repeatable with an interpretive framework. Such a framework constitutes a meta model that defines the elements from which a model may be constructed. For example, if in the model of the data it becomes useful to organize entities into classes, then the meta model would define a class as a modeling entity.

It is important to note that a meta model is just a model of a model. As such, the meta model can be expressed using the same representation as the model. A further implication is that the same information expression may, in one context, serve as a model and, in another context, serve as a meta model

The interpretive framework expressed by the meta model constrains the ways in which data may be interpreted. Clearly, some frameworks offer greater insight than others because their constraints organize and partition the data differently. The power of a given framework depends on the nature of the questions to be asked of the information model. Fortunately, graph-based information representation allows one to apply multiple frameworks to a single body of data - in effect, superimposing multiple interpretations on the data.

5.3 Information Management

Information management, then, consists of creating, maintaining, and interacting with information models. A practical system would, of course, be layered atop a data management system so that the physical representations of the information models could be managed as discrete objects. But such "plumbing" provides only the necessary foundation. The substance of the task of managing information is interacting with the models: expanding the information model by adding new information; exploring the network of relationships to discover implicit information that can be made explicit by adding new relationships to the network; and anticipating new configurations and testing interpretations by simulating the dynamics of the relationships.

6 CONCLUSIONS

The time is ripe for a shift from storage/data-centric to information-centric management of business processes and applications. New approaches and novel application of some historical approaches to information representation and also information management provide the missing enablers. The paradigm shift will not displace data management any more than data management displaced storage management. The new paradigm is an emerging market space that overlays the previous market spaces and can make the business of making decisions orders of magnitude easier and more precise.

7 FUTURE RESEARCH

Category Theory will be studied in relation to Concept Modeling. Category Theory is a mathematical methodology that allows abstraction of data objects in the context of relations to each other and therefore is being pursued as the next viable step in evolving solutions to business management problems. (Hofstede, 1997) However, there is concern that this theory has limited application to the general solution since categorization per se implies imposing a rigid structural solution to a very fluid domain.

Therefore a new approach to representing data nodes containing information and relationships between such nodes called ‘Concept Modeling’ will be investigated as the natural extension where categorization tends to stall. (Gogolla, 1996)

REFERENCES

Chalfant, R., 2005. Achieving Storage Efficiency – the Point of ILM. A StorageTek white paper August, 2005

Clegg, P., 2005 New Graph Technology Eliminates Data Analysis Barriers. A Cogito white paper.

Codd, E. F. 1970. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM 13 (6), 377-387.

Ebert, J., Winter, A., Dahm, P., Franzke, A., Suttenbach, R. 1996. Graph Based Modeling and Implementation with EER/GRAL. In: B. Thalheim [Ed.]; 15th International Conference on Conceptual Modeling (ER'96), Lecture Notes In Computer Science, Proceedings, LNCS 1157. Berlin: Springer.

Gogolla, M. 1996 Towards Object Visualization by Conceptual Graphs. Proc. 4th Int. Conference on Conceptual Graphs (ICCS’96) G. Ellis (Ed.), University of New South Wales, Sydney.

Hofstede, A.H.M ter, Lippe, E. and van der Weide, Th. P. 1997 Applications of a categorical framework for conceptual data modeling. Acta Informatica, v. 34 n. 12, pp 927-963.

Lee, C., Parker, D.S., 2001. Inverting the Database. Proceeding of the 27th VLDB Conference, Roma, Italy.

Milligan, C.A., Halladay, S.H., Knowledge vs. Intelligence. IPSI proceedings IPSI Montenegro-2005, Sept , 2005.

Sowa, J.F., 2002. Semantic Networks http://www.jfsowa.com/pubs/semnet.htm.

UML (Universal Modeling Language) site. http://www-306.ibm.com/software/rational/uml/

Smith, J., 1998. *The book*, The publishing company. London, 2nd edition.