Part I: The Critical Importance of Data Integrity

The Forensic Investigations of Non-Structured Data

Unstructured data is now huge in any company: it's data that is found in applications like websites, documents and emails, and, for many companies, the size is now approaching 85% of all a company's data. Unstructured data is attractive to use as it is more accessible and changeable than the massaged, sanitized, and structured version. Take this unstructured data growing trend and combine it with increased number of litigious incidents among companies, and one can see that more and more companies are becoming very challenged in providing proper legal electronic evidence. Understanding and proving when a company's data is in fact, intact and can stand the test of legal discovery has reached critical mass.

Example of Using an Audio Recording

Software and service options to record meetings such as conference calls is becoming more accessible in the workplace. After decades of research and development, speech recognition technology continues to improve and with it the ability to translate a meeting into text for later review or reference. To take a conference call, record it and then automatically transcribe it is of great appeal for efficient archive, search and retrieval….

However, if an audio recording is submitted as part of a legal case, each recording requires a stenographer to listen to the original recording, and through the affidavit process, provide witness to the whole submitted transcription. The reason automatic speech-to-text recordings won't hold up "in court" is because there is no known voice recognition technology that can absolutely duplicate each phrase, word or expression provided and catch all nuances, whether they are made via various language dialects or through differing voice inflections. In legal investigations, evidence must be shown that will absolutely give witness to the exactness of what was said.

The same principal of witness is required with electronic data. All data is required to hold up to the same standard of being presented in its original binary image state. The challenge for each business is to examine the integrity of its data. When it's absolutely necessary to find the original document for eDiscovery, will a forensic data investigation give evidence that its original state is still intact? Will the evidence also show a process to prove that that document is the only document and could not be preempted by another?

POINTS OF INTEREST:

- Not all data is "stored equally". Trusting your software provider to maintain the integrity of the data is paramount.
- If data is not maintained in its exact format, it's authenticity, accuracy and legal positioning can be questioned. It may not considered legal evidence.

THE REST OF THIS BRIEF:

- The integrity of data is central to the architecture of the *ECM Library* repository.
- Reduction search combined with extensible search capabilities ensure requested data is retrieved.
- Audit of the storage, search and retrieval is critical as part of being compliant to the FRCP process.
- Part II: *ECM Library* strictly adheres to Data Quality rules in design and performance.

Foundational Data Methodology used with ECM Library

Entity Integrity:

Unique Keys and Binary Images: In database design, the “unique key” or “primary key” is required to identify the “entity integrity”, or the case of non-structured data, the file or document that is being archived. In other words, there needs to be one central, primary key that is unique to that file and used as a key identifier. As with all basic database designs, one would assume this key to be in place. However, even if it is in place, if within the file even one byte can be shown to be different since the time of archival, it can be thrown out of a legal case due to its in-exactness.

To provide a more thorough methodology, *ECM Library* goes beyond the industry-standard method of identifying entity uniqueness and integrity: *ECM Library* records ***an exact “Binary Image” of every file***. A binary image is a bit-by-bit representation of the file archived. At this time of publication, the architects of *ECM Library* are not aware of anyone else in the industry who claims they can archive a “Binary Image” and provide this level of data quality. That “original binary image” is restored when a user searches and retrieves any archived file.

Assigning Unique Attributes to Emails: Emails, which by their nature are not associated with a guaranteed unique identifier. In order to compensate for lack of unique key, ***ECM Library assigns each email its own GUID*** (Globally Unique Identifier) at the time or archival. Also, all attributes associated with an email are captured, stored and utilized. *ECM Library* captures where an email was stored (in which folder structure), on what specific machine it was stored or read, and when an email was created, sent and received – by whom and to whom.

All aspects of an email’s associations with owner, receivers, transmission, reception, attachments, folders and dates are required for discovery processes to ensure that:

1) the exact email file has been preserved
2) intent from a legal perspective is included to complete the evidence presentation during the discovery process.

ECM Library captures a binary image of each email and stores this email image and all of its attachments along with its attributes. Critical information about that email including its associations are recorded: its unique identifier plus all associated relative data and metadata. Equally important to the data integrity is the ability to find the data: all emails and email attachments, even including ZIP files and the files contained within are fully indexed and text-searchable.

Issues associated with sole use exact file size. Keeping track of the exact file size archived vs. restored can be misleading due to the trailing byte entries made at times by the database; such as variable length strings. An audit that checks the exact size of a file archived vs. the file size restored can be fraught with errors in interpretation due to the database entries added to file properties. Therefore, although this forensic methodology may be somewhat useful, in itself it is not a conclusive measurement. It does not account for the pure “binary image” that is contained within the data wrapped around the image so the DBMS can store it properly.

Process Integrity

CRCs: Some technologies have suggested using Cyclic Redundancy Checks (CRC) as a means to check data integrity in the archival process. It is often falsely assumed that when a message and its CRC are received from an open channel and the CRC matches the message's calculated CRC then the message cannot have been altered in transit. The danger is that both could have been changed, such that the new CRC would match the new message. CRCs are not, by themselves, suitable for protecting against intentional alteration of data (for example, in authentication applications for data security), because their convenient mathematical properties make it easy to compute the CRC adjustment required to match any given change to the data.

Therefore, CRCs continue to be a helpful tool to verify correctness of transmission but not integrity of archival. *ECM Library* uses and records CRCs, not to ensure the data is an exact replicate, but to ensure the transmission across networks was exacting.

ETLs: Another aspect of process integrity is the methods used for extraction, transformation and load (also known as ETL). *ECM Library* incorporates ETL processing through the utilization of Microsoft SSIS, such that *ECM Library* becomes a warehouse of data that is fully indexed and fully searchable.

Audits and Safeguards: Companies take on risks with any increased level of responsibility and privileges that an employee is given. IT Administrators have special privileges that if not done correctly can create data chaos. *ECM Library* is designed to account for the human error factor so that actions can be tracked and monitored. The user interface provides helpful software decision processes and warnings safeguards against administrative errors. At the user and global user level, providing enough privileges but with monitoring, ensures that there is visibility into user actions, allowing for better isolation of errors. For example, *ECM Library* captures and permanently stores all exact phrases and words captured on a search, the query issued to the database to return the search, which person, which machine, how many rows, which rows , and time of execution and return—are kept in a running log.

Referential Integrity

ECM Library imposes referential integrity throughout its object oriented architecture. In simple terms, all keys and attributes and data are always referenced to a primary key value and cannot exist without this referential link. For example, an attribute can't be deleted if it's tied to a document; a document cannot exist without an owner, an owner can't be deleted if the owner has documents. (However, owners can be made inactive or an administrator can transfer ownership to a new owner.) What differentiates *ECM Library* in the content software industry is that this **referential integrity is defined at the *repository level and ensures imposed RI for unstructured data***. *ECM Library* takes it a step further and exploits these RI Indices to gain unheard-of speed for search and retrieval.

Conclusion

DMA Chicago architects have been in the business of database design, and content management for decades, and have been relied upon by many Fortune 100 companies to implement the most critical of applications. Every part of this extensive experience contributes towards a software platform that will exceed the scrutiny of the sharpest IT questions and concerns. *ECM Library* has been built on a design that exceeds Data Integrity expectations for managing an enterprise's non-structured content.

Part II: The Data Quality Rules

The remaining portion of this Product Brief will discuss in detail types and characteristics of data. This information provides a beginning overview to understand how data can be classified in an archive. There are four categories of data quality rules:

1. The first category contains rules about business objects or business entities.
2. The second category contains rules about data elements or business attributes.
3. The third category of rules pertains to various types of dependencies between business entities or business attributes, and
4. The fourth category relates to data validity rules.

ECM Library adheres to these rules of design.

1. Business Entities

Business Entities are subject to three data quality rules: uniqueness, cardinality, and optionality. These rules have the following properties:

Uniqueness— There are four basic rules to business entity uniqueness:

- Every instance of a business entity has its own unique identifier. This is equivalent to saying that every record must have a unique primary key.
- In addition to being unique, the identifier must always be known. This is equivalent to saying that a primary key can never be NULL.
- Rule number three applies only to composite or concatenated keys. A composite key is a unique identifier that consists of more than one business attribute. This is equivalent to saying that a primary key is made up of several columns. The rule states that a unique identifier must be minimal. This means the identifier can consist only of the minimum number of columns it takes to make each value unique—no more, no less.
- The fourth rule also applies to composite keys only. It declares that one, many, or all business attributes comprising the unique identifier can be a data relationship between two business entities. This is equivalent to saying that a composite primary key can contain one or more foreign keys.

Cardinality— Cardinality refers to the degree of a relationship, that is, the number of times one business entity can be related to another. There are only three types of cardinality possible. The "correct" cardinality in every situation depends completely on the definition of your business entities and the business rules governing those entities. You have three choices for cardinality:

- One-to-one cardinality means that a business entity can be related to another business entity once and only once in both directions. For example, a man is married to one and only one woman at one time, and in reverse, a woman is married to one and only one man at one time, at least in most parts of the world.

- One-to-many (or many-to-one) cardinality means that a business entity can be related to another business entity many times, but the second business entity can be related to the first only once. For example, a school is attended by many children, but each child attends one and only one school.

- Many-to-many cardinality means that a business entity can be related to another business entity many times in both directions. For example, an adult supports many children, and each child is supported by many adults (in the case of a mother and father supporting a son and a daughter).

Optionality— Optionality is a type of cardinality, but instead of specifying the maximum number of times two business entities can be related, it identifies the minimum number of times they can be related. There are only two options: either two business entities must be related at least once (mandatory relationship) or they don't have to be related (optional relationship). Optionality rules are sometimes called reference rules because they are implemented in relational databases as the referential integrity rules: cascade, restrict, and nullify. Optionality has a total of five rules; the first three apply to the degree of the relationship:

- One-to-one optionality means that two business entities are tightly coupled. If an instance of one entity exists, then it must be related to at least one instance of the second entity. Conversely, if an instance of the second entity exists, it must be related to at least one instance of the first. For example, a store must offer at least one product, and in reverse, if a product exists, it must be offered through at least one store.

- One-to-zero (or zero-to-one) optionality means that one business entity has a mandatory relationship to another business entity, but the second entity does not require a relationship back to the first. For example, a customer has purchased at least one product (or he wouldn't be a customer on the database), but conversely, a product may exist that has not yet been purchased by any customer.

- Zero-to-zero optionality indicates a completely optional relationship between two business entities in both directions. For example, the department of motor vehicles issues drivers licenses and car licenses. A recently licensed driver may be related to a recently licensed car and vice versa, but this relationship is not mandatory in either direction.

- Every instance of an entity that is being referenced by another entity in the relationship must exist. This is equivalent to saying that when a relationship is instantiated through a foreign key, the referenced row with the same primary key must exist in the other table. For example, if a child attends a school and the school number is the foreign key on the CHILD table, then the same school number must exist as the primary key on the SCHOOL table.

- The reference attribute does not have to be known when an optional relationship is not instantiated. This is equivalent to saying that the foreign key can be NULL on an optional relationship.

2. Business Attribute Rules

Business attributes are subject to two data quality rules, not counting dependency and validity rules. The two rules are data inheritance and data domains:

Data inheritance— The inheritance rule applies only to supertypes and subtypes. Business entities can be of a generalized type called a supertype, or they can be of a specialized type called a subtype. For example, ACCOUNT is a supertype entity, whereas CHECKING ACCOUNT and SAVINGS ACCOUNT are two subtype entities of ACCOUNT. There are three data inheritance rules:

- All generalized business attributes of the supertype are inherited by all subtypes. In other words, data elements that apply to all subtypes are stored in the supertype and are automatically applicable to all subtypes. For example, the data element Account Open Date applies to all types of accounts. It is therefore an attribute of the supertype ACCOUNT and automatically applies to the subtypes CHECKING ACCOUNT and SAVINGS ACCOUNT.

- The unique identifier of the supertype is the same unique identifier of its subtypes. This is equivalent to saying that the primary key is the same for the supertype and its subtypes. For example, the account number of a person's checking account is the same account number, regardless of whether it identifies the supertype ACCOUNT or the subtype CHECKING ACCOUNT.
- All business attributes of a subtype must be unique to that subtype only. For example, the data element Interest Rate is applicable to savings accounts, but not checking accounts, and must therefore reside on the subtype SAVINGS ACCOUNT. If the checking accounts were interest bearing, then a new layer of generalization would have to be introduced to separate interest-bearing from noninterest-bearing accounts.

Data domains— Domains refer to a set of allowable values. For structured data, this can be any of the following:

- A list of values, such as the 50 U.S. state codes (AL … WY)
- A range of values (between 1 and 100)
- A constraint on values (less than 130)
- A set of allowable characters (a … z, 0 … 9, $, &, =)
- A pattern, such as a date (CCYY/MM/DD)

Data domain rules for unstructured data are much more difficult to determine and have to include meta-tags to be properly associated with any corresponding structured data.

3. Data Dependency Rules

The data dependency rules apply to data relationships between two or more business entities as well as to business attributes. There are seven data dependency rules: three for entity relationships and four for attributes:

Entity-relationship dependency— The three entity-relationship dependency rules are:

- The existence of a data relationship depends on the state (condition) of another entity that participates in the relationship. For example, orders cannot be placed for a customer whose status is "delinquent."
- The existence of one data relationship mandates that another data relationship also exists. For example, when an order is placed by a customer, then a salesperson also must be associated with that order.
- The existence of one data relationship prohibits the existence of another data relationship. For example, an employee who is assigned to a project cannot be enrolled in a training program.

Attribute dependency— The four attribute dependency rules are:

- The value of one business attribute depends on the state (condition) of the entity in which the attributes exist. For example, when the status of a loan is "funded," the value of Loan Amount must be greater than ZERO and the value of Funding Date must not be NULL. The correct value of one attribute depends on, or is derived from, the values of two or more other attributes. For example, the value of Pay Amount must equal Hours Worked multiplied by Hourly Pay Rate.
- The allowable value of one attribute is constrained by the value of one or more other attributes in the same business entity or in a different but related business entity. For example, when Loan Type Code is "ARM4" and the Funding Date is prior to 20010101, then the Ceiling Interest Rate cannot exceed the Floor Interest Rate by more than 6 percent.
- The existence of one attribute value prohibits the existence of another attribute value in the same business entity or in a different but related business entity. For example, when the Monthly Salary Amount is greater than ZERO, then the Commission Rate must be NULL.

4. Data Validity Rules

Data validity rules govern the quality of data values, also known as data domains. There are six validity rules to consider:

Data completeness— The data completeness rule comes in four flavors:

- Entity completeness requires that all instances exist for all business entities. In other words, all records or rows are present.
- Relationship completeness refers to the condition that referential integrity exists among all referenced business entities.
- Attribute completeness states that all business attributes for each business entity exist. In other words, all columns are present.
- Domain completeness demands that all business attributes contain allowable values and that NULL values can be differentiated from missing values.

Data correctness— This rule requires that all data values for a business attribute must be correct and representative of the attribute's:

- Definition (the values must reflect the intended meaning of the attribute)
- Specific individual domains (list of valid values)
- Applicable business rules
- Supertype inheritance (if applicable)
- Identity rule (primary keys)

Data accuracy— This rule states that all data values for a business attribute must be accurate in terms of the attribute's dependency rules and its state in the real world.

Data precision— This rule specifies that all data values for a business attribute must be as precise as required by the attribute's:

- Business requirements
- Business rules
- Intended meaning
- Intended usage
- Precision in the real world

Data uniqueness— There are five aspects to the data uniqueness rule:

- Every business entity instance must be unique, which means no duplicate records or rows.
- Every business entity must have only one unique identifier, which means no duplicate primary keys.
- Every business attribute must have only one unique definition, which means there are no homonyms.
- Every business attribute must have only one unique name, which means there are no synonyms.
- Every business attribute must have only one unique domain, which means there are no overloaded columns. An overloaded column is a column that is used for more than one purpose. For example, a Customer Type Code has the values A, B, C, D, E, F, where A, B, and C describe a type of customer (for example, a corporation, partnership, or individual), but D, E, and F describe a type of shipping method (for example, USPS, FedEx, or UPS). In this case, the attribute Customer Type Code is overloaded because it is used for two different purposes.

Data consistency— Use the following two rules to enforce data consistency:

- The data values for a business attribute must be consistent when the attribute is duplicated for performance reasons or when it is stored redundantly for any other reason, such as special timeliness requirements or data distribution issues. Data should never be stored redundantly because of departmental politics, or because you don't trust the data from another user, or because you have some other control issues.
- The duplicated data values of a business attribute must be based on the same domain (allowable values) and on the same data quality rules.