BUILDING THE ENTERPRISE METADATA REPOSITORY
By W H Inmon

Metadata has been around for as long as there have been programs and files. And while a shop is small and decentralized, there is no recognition of the need for the management of metadata. In addition, as long as all systems are new and well documented, the need for metadata management goes unrecognized.

But in today’s world, systems are not new, are not well documented, and are not all situated in a single small location. Today’s systems are complex and run over many environments. Today’s systems encompass many kinds of technologies with many kinds of users spread over many locations. And most of the systems have little or no documentation, much less up to date accurate documentation. To further complicate matters, the end user has taken over much of the domain of information processing with their own technology. And that technology is frightfully undocumented.

In a word, there has never been a greater need for metadata management across the enterprise.

WHERE ENTERPRISE METADATA RESIDES
Fig 1 shows the many types of places where information processing occurs.

The world of modern information processing is distributed

Fig 1
And in every place where information processing occurs there is found metadata, as seen in Fig 2.

Metadata is found in OLAP processing. Metadata is found in ETL processing. Metadata is found in the data warehouse. Metadata is found in reports. In a word, everywhere there is processing, there is metadata.

WHERE METADATA IS CONTROLLED
Furthermore, in every place where metadata is found, the metadata is housed and manipulated there. People using DB2 for a data warehouse have their own metadata for the data warehouse. People that use the OLAP environment have their own metadata and their own universes. People that archive data have metadata stored as an adjunct to the archive environment, and so forth. The update, creation, and deletion of detailed metadata occurs at the “local” level throughout the enterprise. Fig 3 shows where the control of metadata takes place.
Not only does metadata management occur locally, but the metadata is housed in different technology everywhere it is found. In the data warehouse metadata is stored in Teradata or Oracle facilities. In the OLAP environment, metadata is stored in a Business Object Universe. In the ETL environment metadata is stored as part of Ascential or Informatica. In the legacy environment, metadata is stored as a VSAM directory, and so forth.

METADATA IN DIFFERENT TECHNOLOGIES
Fig 4 shows that metadata is stored in many different technologies across the enterprise.

THE ENTERPRISE VIEW OF METADATA
Certainly local control of metadata is necessary and important. But there is another view of metadata that is important as well. That view is at the enterprise level, as seen in Fig 5.
THE ENTERPRISE METADATA REPOSITORY
In Fig 5 it is seen that metadata is gathered from many different sources and is collected in a single place. The place where the metadata is collected on an enterprise wide basis is called the EMR or enterprise metadata repository. Once populated, the EMR is capable of fulfilling many goals –

- determining where there is overlap,
- determining where there are holes in systems,
- determining the flow of data across the enterprise,
- forming a basis for impact analysis, and so forth.

In fact there are many reasons why organizations want an up to date, accurate EMR.

FOUNDATION SOFTWARE
It is Foundation software specially built for the purpose of enterprise metadata consolidation that fills the information needs for EMR metadata. Foundation software allows enterprise metadata to be gathered, edited, and restructured into a proper EMR.

THE STRUCTURE OF THE EMR
The first step in the building of the EMR is to determine the structure of the EMR. The EMR consists of tables that are defined by the data analyst. Fig 6 shows the tables that can be specified by the data analyst.
The enterprise metadata repository is made up of tables of metadata text.

Fig 6

The data analyst can create as few or as many tables to be placed in the EMR as desired. Fig 7 shows one possibility out of many possibilities.

Fig 7

In Fig 7 the data analyst has specified that there be separate metadata tables for –

Source systems (legacy systems)
OLAP systems
Archival data
Data warehouse
ETL
Data marts.

These tables will end up containing metadata that comes from the metadata sources around the enterprise.

Once these EMR tables have been built, they can be housed in any standard relational technology such as DB2, Oracle, Teradata, or SQL Server.

POPULATING THE TABLES
Once the data analyst has determined the structure of the EMR, the next step is to populate the tables. The tables are populated by using Foundation software. Fig 8 shows that Foundation is used to read and capture metadata from around the enterprise then to place the metadata into the EMR.

Fig 8 shows the role of Foundation software.

Foundation software has three components. Those components are –

- the access of metadata
- the editing of metadata
- the structuring and delivery of metadata to the EMR.

Fig 9 shows the three basic components of Foundation software.

THE ACCESS OF LOCAL METADATA
The access of metadata is not very complicated. Fig 10 shows the kinds of data that can be read for processing inside Foundation.
Foundation software can access a wide variety of text and unstructured data Fig 10

It is seen that Foundation is capable of reading and accessing a wide variety of metadata.

EDITING FUNCTIONS
Once the data is accessed, the data is then edited. Fig 11 shows some of the basic functions of the editing process.

The functions that are done by Foundations metadata editor are shown by Fig 11. Each of the functions will be explained separately.
REMOVING BLOCKS OF LOCAL METADATA TEXT
One of the functions of the Foundation 5 editor is to remove large blocks of text. Fig 12 shows this function.

In Fig 12 it is seen that metadata text has been taken from the local metadata environment by Foundation. There are large blocks of text that are stripped away from the source. Upon stripping away the text, large blocks of text can be deleted. In this case, the text beginning “Oracle Directory…” and ending with “…Acme Hardware” is deleted.

ADDING BLOCKS OF TEXT
In the same vein Foundation editor can add text as seen in Fig 13
Fig 13 shows that text has been added to the local metadata text that has been read. Both the deletion and addition of text is done manually.

In Fig 13 the metadata text that has been added includes “inventory table”, “inventory part number” and so forth.

**SEPARATE METADATA ENTITIES**
In addition to adding and removing text, the data analyst can select blocks of text and treat them as separate entities. Fig 14 shows such a treatment.

Fig 14 shows that metadata text from the local metadata environment can be read and isolated. Fig 14 shows that three blocks of text have been isolated – Table ABC, Table BCD, and Table CDE.
DIFFERENT DESTINATIONS
Once the blocks of text have been isolated, they can then be sent to different destinations. Fig 15 shows that Table BCD is sent to the table containing data warehouse metadata, Table CDE is sent to the table containing OLAP metadata, and Table ABC is sent to the table containing the archival metadata.

4 - place text in different destinations

Fig 15

ADDINNG THE DATE OF PROCESSING
Another simple editing feature is the addition of the date of processing to the output metadata, as seen in Fig 16.
ADDING THE SOURCE OF LOCAL METADATA

And another simple editing feature is the addition of the original source of metadata, as seen in Fig 17.
ADDING POINTERS FROM ONE BLOCK TO ANOTHER

Pointers from one block of text to another can be created, as seen in Fig 18.

These pointers are created manually. These pointers indicate a relationship from one block of metadata to another block. Typical of the relationships that are indicated are sources and targets. The pointer can be used to point from a source to a target. In doing so a chain of analysis can be created for impact analysis.

AUDITING POINTERS

Fig 19 shows that an audit can be done to ensure that the pointer is not directed to a destination that does not exist.
SYNCHRONIZING UPDATES FROM THE SOURCE TO THE EMR
Yet another feature of the Foundation software is the detection of when an update to the EMR needs to occur. Fig 20 shows that Foundation software is sensitive to changes in the source metadata.
Fig 20 shows that an update has been done from a source on March 16. Now, on December 18, the system is asked to make another update. When the system is asked to make another update, the system senses that no changes have been made to the source since the last update. Therefore the system makes no updates since no source system data needs to be updated.

THE FINAL OUTPUT
The final output of the EMR is shown in Fig 21.

Fig 21 shows that the EMR consists of tables that have been defined by the data analyst. The tables can be related by means of pointers. Each table contains blocks of metadata text that have passed through the Foundation editor. Each block of text contains the date the text was last edited and the source of the text.

The output tables can be placed in any standard technology such as DB2, Teradata, SQL Server, or Oracle.

About Inmon Data Systems

Founded in Colorado by Bill Inmon, Guy Hildebrand and Dan Meers, Inmon Data Systems (IDS) is a software company dedicated to the proposition that there needs to be a
bridge between the worlds of structured data and unstructured data. IDS has foundation technology that allows unstructured data to be brought into the structured environment and once there, integrated into the structured environment.

Applications – unstructured visualization (with Compudigm)
  - Enterprise metadata consolidation
  - CRM enhancement
  - Communication compliance
  - Email and unstructured indexing for bulk storage

IDS is located in Castle Rock, Colorado.

Contact Carol Renne at 303-973-3788 for further information.