# A PROPOSED APPROACH TO METADATA MODELING

By
Jinane Hashem

A thesis
Submitted for partial fulfillment
Of the requirements for the degree of
Masters in Computer Information Systems

Department of Computer Science
Faculty of Natural and Applied Sciences
Notre Dame University



November 2005

# A PROPOSED APPROACH TO METADATA MODELING

**By**
**Jinane N. Hashem**

**Approved**

Dr. Marie Khair: Associate Professor of Computer Science
Advisor

Dr. Khaldoun Khaldi: Assistant Professor of Computer Science
Member of the Committee

Dr. Mario Missakian: Assitant Professor of Computer Science
Member of the Committee

Dr. Melhab Keyrouz: Associate Professor of Mathematics
Member of the Committee

Date of Thesis Defense: November 21st, 2005

# Acknowledgements

This study is dedicated to my family and friends, for their steady support during my sojourn into the excitement of the world of metadata.

Special thanks are due on one hand to my dearest university NDU and all the staff members - and most specifically to Dr. Marie Khair for her guidance and counsel, and on the other hand to my company SoftSolutions for all the help and encouragement offered - and most particularly to Dr. Marie Aboujaoude to whom I would like to express a great deal of gratitude.

With a hope that this will be the opening gate to several other future efforts and studies...

# Abstract

Metadata is considered a very useful and valuable component in handling data and delivering decisional information, in managing contents on the Internet, in databases and warehouses, in enabling the finding of relevant information, and in effectively providing a multiplicity of useful services.

Data, in point of fact, can be metaphorically compared to a dead substance; and all acknowledge the fact that data, by merely existing in systems, represents almost nothing. However, it is the awareness of the exact length and value of this data that builds the valuable expensive information.

This thesis presents a detailed overview of metadata and the existing approaches used in metadata modeling, then proposes a new approach rather deviated from the common standards and Web approaches already used; while the previous ones address specific problems, the quest in this thesis goes for a global approach that would suit the largest possible set of applications and scenarios, help in discovering this magic recipe of turning lifeless uninteresting data into valuable active metadata information, and by this define a grand unified metamodel that can be called an Information Model. Thus, having tackled different sides of the available metadata approaches, the objective develops into obtaining - in addition to a comparative study between the existing and the proposed approach - several advantages of the latter one, beginning with a representation of its step-by-step metadata construction methodology, its integration of web and applications metadata, the incorporation of business rules into it, the use of the Einstein Simplicity Theory of dividing and regrouping elements during its conception, and finally ending up with a detailed example to support it, and following the proposed procedure in its implementation.

# List of Figures

# List of Tables

# Table of Contents

# CHAPTER I

# PROBLEM DEFINITION

# 1. GENERAL OVERVIEW OF THE PROBLEM:

With the awareness of the fact that metadata means different things to different people according to their specific needs, the approaches considered so far for metadata modeling generally deal with each need separately. For instance, most modeling standards for Web resources handle search utilities without addressing other problems and potential requirements; some governmental institutes, and on the other hand various working groups, have so far developed metadata that fits their exact necessities; and a few companies use the system metadata to meet their precise application or integration needs.
Due to all these facts, professionals wind up with somehow limited metadata models, which naturally imply limits in the potential power, value and efficiency metadata can normally offer.

In addition to the above, metadata solutions can also pose significant cultural, technical and financial challenges, challenges that drive the less informed people about their efficacy to use them as little as possible...

Moreover, and especially when it comes to the business rules embedding to the core metadata, some employees discreetly tend to reserve or monopolize the business rules, and thus these rules remain hidden in the analyst or even the programmer's mind, causing a certain dependence or reliance on a few business knowledgeable people, which without any doubt sometimes turns into something similar to enslavement that every CEO or company owner fears; not to mention the future of the whole company and the fate of the rest of the employees if one day one or more of the business wise well-informed people decided to quit...

Recently, in an EContent Magazine article "To Metadata or Not To Metadata", published in late October 2004, the following was clearly stated: "A couple of things have become increasingly clear: Metadata is not going away and there is no one simple solution to how to add metadata and maximize its value. [24]. These issues are constantly being explored in any Metadata Workshop (for example, the DCMI –Dublin Core Metadata Initiative - 2004 Workshop) [4]. Even more, in both his "Metadata Evolution Management" January 2005, and "Know How to Manage Your Metadata" February 2005 articles, Jim Gabriel, an architect, author, entrepreneur, inventor, and XML specialist, clearly stated that "the disadvantages of any metadata-driven application environment are due entirely to the limitations of metadata in general and XML schemas in particular" [11,12].

## 2. PROBLEM STATEMENT:

All in all, the general problems witnessed in traditional metadata approaches can be summed up in the fact that they:

- Commonly deal with restrictive sets of needs
- Represent technical, financial and cultural challenges that drive the less informed people away from using metadata in general
- Are not comprehensive and clearly stated in an instructive or didactic way
- Generally lack business rules embedding and by this imply a whole set of technical and political problems in a company
- Have limitations that restraint their efficiency and advantages and minimize their value

## 3. INTRODUCTION TO THE SOLUTION AND THESIS ORGANIZATION:

That said, the reader becomes aware of most kinds of problems the Internet and the whole computer community suffers from on a daily basis. The solution to be proposed by this study shall therefore present one complete inclusive comprehensive approach that meets at the same time the widest set of needs and metadata utilities, and that is clearly defined in a visible step by step algorithm or procession method, in a way that it additionally fits to become a didactic-like approach.

The task will be carried out throughout the five chapters of the thesis, starting with the actual one which defines the problem; the second chapter provides a detailed definition of metadata, its different types, usages and users, advantages, disadvantages and applications. While the third chapter identifies available metadata modeling approaches and standards, their different properties and specifications, the fourth chapter presents the proposed approach, its step by step algorithm, its conceptual and physical model and data dictionary, along with a detailed illustrating case study, and eventually its advantages and disadvantages, as well as a comparison with existing approaches. At the end, the fifth chapter comprises the conclusion of the thesis and introduces in general its possible future subsequent corresponding work.

# CHAPTER II

# INTRODUCTION TO METADATA

# 1. INTRODUCTION:

Each and every one absolutely agrees that it is not in any way possible to go into a huge gigantic library, containing millions of books without labels or subtitles, without identifying cards... In the same context, surfing the web without the help of the search engines is also unfeasible. In other words, had it been required to act in such environments without the metadata being available, one would have found himself doing an impossible mission, with for instance the only way to proceed the fact of randomly guessing the correct URL, or opening the required book, on every time a piece of information is sought, infinitely, by trial and error...

Perhaps these could be practical examples that help in picturing one of the advantages provided by a well designed metadata model, before exploring the rest of the multiplicity of advantages, and catching a glimpse of how metadata adds life and value to all the lifeless data available in all systems and servers.

One thing to be kept in mind, however, is the fact that in the future, when the metadata languages and engines are more developed, metadata will also, without any doubt, form a strong basis for a web of machine understandable information about anything: about people, things, concepts and ideas.

In this chapter, the detailed structured definition of metadata - whether for system and applications metadata, or for digital and Web metadata – as well as its use and interest for both technical and business users will be explored. Next, metadata costs and disadvantages, followed by their applications and benefits will be stated, before concluding with the notion of the all-time information equation that relies a great deal on metadata quality, and by this increase the awareness of how much essential and imperative the call for a carefully designed metadata repository henceforth shall be.

# 2. METADATA DEFINITION:

## 2.1 Introduction:

The great dilemma that lies behind the metadata concept is the fact that it is often variously interpreted according to its various usages and needs. Simply speaking, Metadata is described as data about data, or as a database about a database. Metadata consists of assertions about data, and such assertions typically, when represented in computer systems, take the form of a name or type of assertion and a set of parameters, just as in the natural language a sentence takes the form of a verb and a subject, an object, and various clauses.

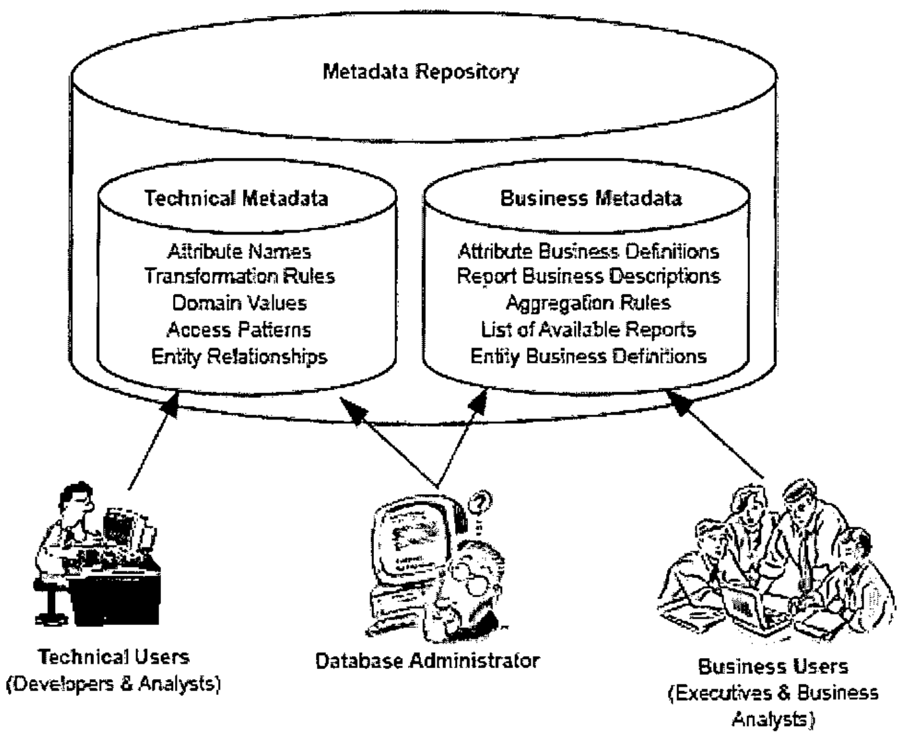


*Figure 1: Common Metadata Types & Users* [27]

## 2.2 Web Metadata:

In the case of Web metadata, machine understandable information about web resources and documents is dealt with. Today, when digital information is being created at a stupendous rate, an efficient and effective way to locate, access, manage, and understand it all is needed – and that's where metadata comes in. Metadata may refer to any resource that has a URI; moreover, it may be stored in any resource, no matter to which resource it refers.

There are ways of representing the above elements, going from messages, labels, specifying labels, statements, and distinguishing between them. Generally speaking, it is always advantageous to acquire as much as possible of the syntax and semantics referencing a document. It must be possible to mix multiple vocabularies within the same scope. The syntax and structure should be such that as many manipulations as possible can be done without having to know the semantics of the vocabulary in use. A common vocabulary for basic logic and knowledge representation functionality will be required [13].

## 2.3 System and Applications Metadata:

When dealing with system and application metadata, the terms *meta database* or simply *metabase* are often used.

Roughly speaking, what is being pointed at in this case consists of the sets of text descriptions about the columns, tables, views, procedures, etc. used in a database. The schema diagrams at the logical level, business rules implemented through various data validation algorithms such as declarative integrity and triggers, sizing algorithms used for the physical database, etc., can all be included in the metadata model; the more rich it can be made, the more powerful it will reveal itself to be.

In a data warehouse metadata, for instance, all of the above data needed for an OLTP database and, in addition, included data about all the mappings needed for extracting data can be found [17].

## 2.4 Metadata for Technical Users:

The role of metadata for technical users can be perceived in the DBA's extracted information from the metadata dealing with sizes of database objects, availability of free space, trend of allocation of next extents to segments, datafile sizes, disk space allocation, etc.

The DBA can utilize the scripts given from the metadata system repositories and views, or they can reverse engineer the logical model from the data warehouse and then use some metadata tool to extract the information.

On the other hand, always in the context of technical metadata users, the developers' needs are different from those of a DBA, and lean more towards the applications using the database or the data warehouse. They may want to know the index structure on the fact and dimension tables, logical versus physical data models, schema diagrams, details of mappings between the source and target data stores, data validation as enforced by the metadata, etc.

## 2.5 Metadata for Non-technical Users:

Metadata takes the business users closer to the raw data by opening a window of understanding for the actual data, the way it is stored and the possible uses of data. This increases the possibility of use of the data and applications built on it, giving a real boost to the business objective called growth. In other words Metadata is an abstraction layer that masks the underlying technologies, making data access friendlier to the business users.

Generally speaking, these users are primarily business analysts who are subject area experts from a business viewpoint, but are not necessarily technically competent. They use the metadata to ensure that the warehouse can adequately provide data to support the DSS of the organization.

Their primary focus is on the schema diagrams, descriptions of data elements at table and view levels, data validation routines enforcing the business rules, verification of source and target for individual data elements, etc. [17].

# 3. METADATA DISADVANTAGES:

The advantages of metadata go way beyond its disadvantages. They can be summarized in the cost of creating and adding metadata, its evolution management at one hand, and continuing on the other hand with the difficulty of doing it well and the associated problem that poor-quality metadata can actually make its presence worse than no metadata at all.

## 3.1 Cost and Evolution Management Factors:

The cost argument will first be tackled. Generally speaking, metadata systems are expensive to build and maintain. On several occasions, and especially during workshops and seminars, many participants have cited their practices in which they testify being charged with several hundreds of thousands of dollars for a full-scale metadata implementation. Needless to say, this can certainly seem like an exorbitant amount of money, especially for a company that is still using for example only some tens of thousands of dollars search engine for its intranet. In addition, this expense is just for adding metadata to a large existing content repository but does not take into account the additional cost of maintaining and adding new metadata [24].

This brings to the metadata evolution management problem, which is the real problem facing the long-term lifecycle management of web services development projects. The disadvantages are due entirely to the limitations of metadata in general and XML schemas in particular. Nowadays schemas must be expected to change. The XML schemas describing web services message payloads are application-specific, bespoke metadata, which requires human involvement when it evolves. Unfortunately, developers modify schema-driven applications by editing the schemas. There is no other way, and there is no robust, scientific mechanism for identifying where every object has been defined and referenced. This traps developers in a manual maintenance exercise. For multiple schema families and multiple developers (or worse, multiple teams of developers), a very serious risk of conflicting modifications and inconsistency exists.

## 3.2 Poor Quality Factor:

Another argument against building metadata is the immense difficulty that many programmers and analysts find in doing it well. The difficulty of effectively employing metadata can easily be seen in the abysmal quality of the metadata associated with the unstructured content found on most corporate intranets. In evaluating corporate intranets, time and again missing metadata fields and elements, missing values from the fields that have been defined, very poor quality values in even simple elements, and inconsistent values among similar objects or documents are encountered. One interesting aspect of bad metadata is that it does not just detract from getting full value from the effort to add metadata. For instance, if the search engines example is considered, and the case when metadata is tweaked to be used like keywords in ranking; if it happens that someone puts in bad metadata, the document might be lost forever, or it will not turn up when someone searches for a relevant search term, but will turn up for an inappropriate search term, and promptly be ignored. The same paradigm will be witnessed in the case of query builder or reporting engines relying on bad quality metadata. Indeed, all these examples constitute a demonstration of how bad and low quality metadata can sometimes be worse than no metadata at all [11, 18].

## 3.3 Conclusion:

Given the rather pathetic record that many metadata efforts have racked up, it is little wonder that some organizations, being unaware of the long term benefits of metadata, and only considering the tangible cost, evolution and quality management factors, have begun to question the entire value of implementing metadata, ignoring the fact that the cost of doing metadata has to be weighed against the cost of not doing metadata...

# 4. METADATA ADVANTAGES:

To the other side of the story, the various advantages offered by metadata are explored. Very broadly, and by being the bridge between business and technical information, metadata can greatly increase the level of productivity and improve the decision-making process of knowledge workers within an organization.

With high pressures for growth and competition, business intelligence is becoming a standard business practice for companies the world over. Without Metadata solutions in place, users will not be able to exploit the database, data warehouse or any business intelligence solution completely and effectively.

Data within a database is just that, data. It becomes information only when it can be effectively extracted and distributed in an understandable form. Using metadata has come a long way; it allows for free-flowing information and is putting the ability to extract information into the hands of end-users.

A company's ability to manage data, information, and knowledge will determine how successful a company can be; or whether or not they can be successful at all. To manage data, information, and knowledge, companies need to know what data they have. Companies need to know precisely how their data is being used and how that data can be used to create competitive advantage. To know these things, a company needs to manage and use its metadata.

That said, the main advantages and factors that have triggered the need for metadata in today's business can be summarized in the following aspects:

- A metadata repository significantly reduces both time and cost of analysis and development by documenting the data transformation rules, data sources, data structures, and context of the data in the Web, databases, data warehouse and data marts. This information is critical because without metadata, the transformation rules often reside only in the IT staff's collective memory. Because the results of the analysis and developmental changes are captured and retained in the metadata repository, the benefits are long lasting, helping to reduce the costs of future releases and the likelihood of developmental errors.

- Metadata provides corporation's IT department with the ability to maintain and grow its system over time. Metadata is vital in understanding what data is used most often and for locating the dormant data (i.e. data in the database that is not accessed) to fine-tune the different systems as they grow. Metadata allows corporations a much greater level of knowledge about their decision support architecture. This information enables companies to make better architectural decisions and add greater flexibility to change the decision support systems.

- Metadata increases business user's confidence in the data that they rely on in the corporate decision support and operational systems. Metadata accomplishes this feat by providing a semantic layer between these systems and the users. In simple terms, metadata looks to translate the technical terminology into business terms that the users are familiar with. For example, when a business user is looking at profitability numbers it would be very valuable to him to understand the formula used to calculate the numbers and what systems were sourced to provide the data for the formula.

- Metadata provides unified view of the data, in other words, one location to see information about the data.
  Data definitions through metadata solve various problems. For users to get value from names, postal codes, addresses, risk scores, or other data elements, these elements should be presented in a standardized, understandable way. Data that means the same thing but is stored in various ways can confound a well-intentioned end user. A company protects its systems by using standard data definitions. For example, database administrators can come and go, but the data definitions will remain meaningful and consistent. Comprehensive metadata definitions mean never having to explain what a given data element really signifies. Metadata also buffers the

system from endless user searches by letting users query metadata to ensure that what they are looking for exists.

Metadata can tell the user which database or table to query [5, 13].

## 5. METADATA APPLICATIONS:

What makes metadata different from ordinary data is how it is used. It is found in documents, messages, images, sound streams, and videos. The term metadata or "data about data" also refers to data that are used to describe a data set, such as the content, quality, and condition of data. It is a set of facts about data and other information elements. It is everything except for the data itself, and it is undeniably important. This importance can be witnessed in the multiplicity of applications and contexts in which it can be used.

Good metadata helps people find the information they are looking for. Searching through unstructured text (i.e. performing a full-text search) or using uncontrolled terminology (i.e. keywords) may yield tens of thousands of results, the majority of which are usually irrelevant to the searcher. The structure of metadata records allows searching for terms in discrete elements (e.g. title, subject). Search results are therefore fewer and more focused.

In various contexts, for example governmental ones, metadata serves separate but interrelated functions simultaneously, such as information sharing and exchange between federal organizations and their partners, identifying appropriate clusters and gateways content, as well as the internal information management needs of federal organizations. In other systems, metadata is the key element to guarantee, track and manage the workflow and actions of the different entities among several applications on one hand, to build queries and reports according to users' specifications on the other. In addition, metadata plays a major role in the integration and the synchronization of data between dissimilar systems.

Metadata is a web or any other system writing skill. It helps readers quickly find what they need. Metadata is an essential part of successful web sales and marketing efforts. It helps influence people to buy. In every sense, metadata has the power to drive action. It can help achieve several objectives [8].

In short, according to the different user needs and business requirements, metadata can present a wide set of additional applications through the various elaborated advantages.

## 6. CONCLUSIVE FACTS:

This chapter will be concluded with two very central inherent extracts - that once again demonstrate the imperativeness of metadata's role in any system or concept - one from the March/April 2005's Oracle Magazine: "Zen and the Art of Information", and the other from an April 2005 issue of Dbazine (The online community for database issues and solutions): "The Power of Metadata".

In the Oracle article, starting from the hypothesis that naturally asserts that data is totally different from information, what is called "The Information Equation" is defined and elaborated as:

*Information = quality (data + metadata)* [7].

In plain English, they go on stating that *information* (human-readable, no-batteries-required information) equals *data* (which computers love to collect and trade) plus *metadata* (which is data about data, or context), with *quality* applied. While advising mathematicians not to try to prove this, but rather accept it at face value, they conclude that in simple terms, the clear distinction between data and information is the application of context metadata and quality.

The Dbazine article stresses on the annual increase of the amount of data being stored exponentially, while there are limited tools to manage and pull all the data together and make some information sense out of it. However, they go on stating that there is hope in the future, with a handful of companies that are effectively bringing this information together and placing it in the hands of business users. Through the power of metadata, corporate information resources can be pulled together; what data there is, where the data resides, can be understood and ultimately information can be placed in the hands of the end-user more quickly. After all, this is where it belongs [10].

To conclude, in today's competitive environment when every company is looking for an edge over its competitors, building a Meta Data Repository is no longer an option, but an absolute requirement for improvements in business. Business community has realized that Metadata is not only crucial for the management of the IT applications but is also necessary for the effective use by the business end user.

# CHAPTER III

# METADATA MODELING APPROACHES

## 1. INTRODUCTION:

During the quest to finally build an Information Warehouse based on qualitative metadata, several metadata design methods and techniques have been encountered. In this chapter, a detailed overview of the main existing modeling standards and structures mostly known so far in the world of metadata modeling will be provided.

In order to offer a Common Information Space, whether on the web or on any other system or application, meta-data is used in various ways. At system level, meta-data attributes may be stored within the HTML code itself by way of so-called MetaTags, or they may be stored in separate files (Object Descriptor Files - ODF) pointing to the actual information object. The attributes captured in these meta-data structures are then provided to information-hungry users or agents [17].

In what follows, the different database server system metadata, the structural RDF model implementations and Dublin Core standard will be overviewed; the advantages and disadvantages of each approach, along with a few examples and implementation applications will also be cited.

## 2. BUILT-IN METADATA:

As a first step, it is essential to overview the system built-in metadata repositories that are commonly used for different purposes. Needless to say, this type of metadata is static and entirely dependent on a certain database's state, therefore it cannot be directly accessed, codified or structured in an instantly usable way neither for a certain application's need, nor – naturally - for the semantic web, since the business add-ups tend to be hard to acquire.
Here is a summary of these elements present at the conceptual internal level of a database no matter what is the server used.

## 2.1 Oracle Metadata:

Oracle's data dictionary tables and views and the V$ dynamic performance views are two examples of built-in metadata. They collect real-time data and statistics about all the database objects, the components of the SGA, background processes, and user transactions among others.

The DBA can utilize the scripts given from the metadata system repositories and views or can reverse engineer the logical model from the data warehouse and then use some metadata tool to extract the information.

Oracle provides static views in the data dictionary, popularly called V$ views. These are mainly used to monitor the performance of both the instance and the database for an application. The X$ tables underlie the V$ views.

Some examples depicting some very well known Oracle data dictionary views include:

- All_all_tables: Description of all objects and relational tables accessible to the user.
- All_objects: All tables, views, synonyms, sequences available in a certain database.
- All_col_comments: Comments on columns of accessible tables and views.
- All_col_privs: Grants on columns for which the user is the grantor, grantee, owner, or an enabled role or Public is the grantee [20].

## 2.2 SQL Server Metadata:

Similarly to the concept explained above, Microsoft SQL Server has its metadata structures, as well as several tools for retrieving metadata.

All told, there are around 100 system tables that can be broken down into seven groups, and several more information schema views based on these tables.

Some critical instances of these structures are:

- Sysfiles: Containing all files of a database
- Syscolumns: Summarizing all tables' columns in a database
- Sysobjects: Stating all database objects
- Systypes: For user-defined datatypes.

- Tables: View containing a row for each table in the current database for which the user has permissions.
- Referential_Constraints: View containing a line for each foreign key constraint in the database [16].

**2.3 System Metadata Conclusions:**

All in all, the same goes for DB2, Progress or any other database server where metadata plays an essential role and tends to be, in their context, most useful to database administrators and application developers, rather than to end users.
This multiplicity of structures, naming conventions, and groupings, as well as the static nature depending on the databases' state as described earlier, limit the usage of the system metadata and make it impossible to use it solely for business and application needs.

## 3. THE DUBLIN CORE STANDARD:

The Dublin Core standard is the leading international metadata standard for on-line resource discovery. It follows a certain web standard and groups the metadata in a structured hierarchical way, often in HTML documents. As a concrete instance, META Tag Generator is a tool that accepts user input and generates HTML code according to the standard that can be pasted into an HTML page.

**3.1 Introduction:**

The Dublin Core metadata element set is a standard for cross-domain information resource description, where an information resource is defined to be "anything that has identity". The Dublin Core project (DC) grew out of a workshop sponsored by OCLC and the National Center for Supercomputing Applications. The Dublin Core metadata is described as an efficient and simple metadata for electronic articles and digital objects. The Dublin core elements describe a journal article, movie, image etc.

## 3.2 Dublin Core HTML Elements:

The Dublin Core Metadata Element Set represents a simple resource description record. Elements are designed to be used by content creators. The Core contains just 15 metadata elements:

- **Subject and keywords:** The topic addressed by the work; "Subject" contains one or more words or phrases (descriptors) selected from an authorized controlled vocabulary to describe the subject of the intellectual content of the resource.
- **Title:** The name of the object;
- **Author or creator:** The person(s) primarily responsible for the intellectual content of the object; "Creator" specifies the name of the organization(s) responsible for creating and maintaining the resource. Depending on local convention, this name may include an organizational hierarchy that includes the responsible unit.
- **Date:** "Date" specifies one of two dates. The first, date created, identifies the date the resource was first posted on the Web. This element is mandatory. The second, date modified, identifies the date a substantially revised version of the resource was posted on the Web. This element is mandatory only when applicable.
- **Language:** Language of the intellectual content;
- **Publisher:** The agent or agency responsible for making the object available;
- **Description:** textual description of content;
- **Other Agent:** The person(s), such as editors and transcribers, who have made other significant intellectual contributions to the work;
- **Object Type:** The genre of the object, such as novel, poem, or dictionary;
- **Form:** The data representation of the object, such as Postscript file or Windows executable file;
- **Identifier:** String or number used to uniquely identify the object;
- **Relation:** Relationship to other objects;
- **Source:** Objects, either print or electronic, from which this object is derived, if applicable;

- **Coverage:** The spatial locations and temporal duration's characteristic of the object;
- **Rights management:** a rights management statement, an identifier that links to a rights management statement, or an identifier that links to a service providing information about rights management for the resource.

Each element is repeatable and optional (however many organizations applying this standard require the first five elements to be mandatory), and the entire set has been defined as extensible.

Each Dublin Core metadata element can also have a sub-type and sub-scheme information.

For example, if an existing scheme is being used for subject and keywords, then this information can also be attached to the element name.

## 3.3 General Guidance for Developing DC Metadata:

As already stated, the majority of agencies require five metadata elements for describing Web resources: Title, Creator, Date, Language and Subject. However, a given department or agency may require a larger element set which may include optional Dublin Core elements, elements from other metadata standards, and/or locally-defined elements. Tools and procedures used to populate these elements will be determined by policies established by the department or agency and by the information system(s) used.

Although metadata information may be input using an electronic form, metadata embedded on an HTML page will look like this:

```
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/">
<meta name="dc.title" content="xyz">
<meta name="dc.creator" content="xyz">
<meta name="dc.date.created" content="xyz">
<meta name="dc.date.modified" content="xyz">
<meta name="dc.language" scheme="ISO639-2" content="xyz">
<meta name="dc.subject" scheme="gccore" content="xyz">
```

(Where *xyz* represents the metadata value).

The preceding template is the same for all Web pages whether pages are in English only, in French only, in both official languages, or in any combination of English, French and any other language.

When a Web page contains text in more than one language, the Title, Creator, Language and Subject elements are repeated to show separate elements for content in each language. It is unnecessary to repeat the Date element as the content is a numeric value. For some elements (Subject, Audience, Coverage, Format and Type), content must be selected from authorized lists of values.

3.3.1 Expressing Dublin Core Metadata Elements:

Metadata can be expressed in different ways, like HTML, XHTML or XML. When expressed using HTML or XHTML syntax, metadata elements can be embedded in the Web resources or they can reside in an external file. These syntaxes generally require that metadata be held in an external file.

3.3.2 Syntax:

In the context of these guidelines, syntax refers to the rules for the construction of metadata elements.

The basic HTML metadata element pattern is: <meta name="abc" content="xyz"> -

where *abc* is the element name value (e.g. "dc.title") and *xyz* is the information pertaining to the element (e.g. the title of a resource).

When the element requires the use of a scheme the pattern is:

<meta name="abc" scheme="def" content="xyz">

where *abc* is the element name value (e.g. "dc.subject"), *def* is the label of a recognized scheme, and *xyz* is the information pertaining to the element (e.g. the terms selected from the controlled vocabulary or scheme). If content from more than one scheme is used, the element must be repeated for each scheme.

In HTML, all parts of the metatag (except the values in the content = attribute) must appear exactly as shown in the examples. Although the syntax appears to be in English, it is actually machine-readable code that should never be translated. Values appearing in the content = attribute will be in English or French for instance, depending on the language of the resource.

An example of a correct HTML syntax for a French resource is:

```
<meta name="dc.title" content="Quoi de neuf">
```

Not:

```
<meta nom="dc.titre" contenu="Quoi de neuf">
```

3.3.3 Use of the Reference Statement:

The following statement is necessary for machine validation of the Dublin Core schema and must appear once in the <head> of every Web resource:

```
<link rel="schema.dc" href="http://eanucc/org/elements/1.1/">
```

When the audience element is used, a second instance of the <link rel> element must be specified:

```
<link rel="schema.dcterms" href="http://purl.org/dc/terms/">
```

3.3.4 Use of Upper and Lower Case:

Dublin Core expressed in HTML (e.g. <meta name="dc.title">) is always in lower case. Content may contain any alphanumeric characters (e.g. upper and lower case, numbers and symbols). However, terms selected from authorized controlled vocabularies must be exactly as they appear in the controlled vocabulary being used, including case and punctuation.

3.3.5 Accented characters:

As is the case with Web document content in general, metadata content can include accented characters (e.g. "é") or character entity equivalents (e.g. "é"). Although using the character entity equivalent provides an extra measure of assurance that the accented character will be widely decipherable, using the accented character itself is usually sufficient since most modern browsers and search engines interpret and display the ISO Latin-1 Character Set. Accented characters or character entity equivalents should be applied consistently across Web sites.

3.3.6 Description vs. Evaluation:

Metadata is used to describe resource content, not to evaluate or comment on that content. It is inappropriate to include editorial comments in metadata.

3.3.7 Uniqueness of Metadata Tagging:

Every Web resource should have its own unique set of metadata tags reflecting the content of that resource. No two Web resources should contain exactly the same metadata.

3.3.8 Populating the Content Value:

The content attribute of any metatag (<content="xyz">) must have a value; it cannot be left blank. Mandatory metadata elements (Title, Creator, Subject, Language, Date) must have content. For optional elements, if a resource does not require an optional element, that element should be omitted from the metadata entirely.

3.3.9 Order of the Elements:

There is no prescribed order for the elements in a metadata record.

3.3.10 Repeatability:

With the exception of <dc.date.created>, all elements are repeatable.

3.3.11 Updating Metadata as Resource Content Changes:

The metadata for a resource is a reflection of the content of the resource. Whenever resource content changes, metadata developers must ensure that metadata still reflects the content accurately.

3.3.12 Authorized sources of terminology:

In order to ensure successful retrieval of Web resources, it is important that the form of name of the originating department or agency be identical in all metadata records. Authorized sources for organizational names must be used.

3.3.13 Organizational Hierarchies:

Local metadata policy should include guidance on the level of organizational detail to be used when expressing <dc.creator>.

While use of the departmental or agency name alone is acceptable practice and is usually sufficient, the addition of organizational levels could be helpful or necessary. For example, lower levels in the organizational hierarchy should be included when a major organization is part of a larger department or agency and would not appear in <dc.creator> at all if the parent organization alone were used.

## 3.3 Sample Metadata Source Code:

The following is an example of how required and optional metadata elements would appear in the HTML source code for an English site. Required elements have been highlighted in bold face.

```
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/">

<link rel="schema.dcterms" href="http://purl.org/dc/terms/">

<meta name="dc.title" content=" Item Main Catalog">

<meta name="dc.creator" content=" Item EAN.UCC Schema Organization, Item Global Attributes Department, Item Identifier Responsible, GTIN Creator ">

<meta name="dc.language" scheme="ISO639-2" content="eng">

<meta name="dc.date.created" content="2004-00-00">

<meta name="dc.date.modified" content="2005-03-17">

<meta name="dc.subject" scheme="itm" content="Item Info; Information header; Informtion Details; Tarif Attributes">

<meta name="dc.subject" scheme="itmcore" content=" EAN.UCC Attributes, WWRE Attributes, TRANSORA Attributes">

<meta name="dcterms.audience" scheme="retailer" content="supplier">

<meta name="dc.coverage.spatial" scheme="itmgeoname" content="USA">

<meta name="dc.description" content="Links to information and communications technologies in global attributes, local attributes, mandatory attributes, and optional ones">

<meta name="description" content=" GTIN; GLN; Target Market ">

<meta name="dc.type" scheme="itmtype" content="resource list">

<meta name="keywords" content=" Item Info, Information header, Informtion Details, Tarif Attributes, EAN.UCC Attributes, WWRE Attributes, TRANSORA Attributes, GTIN, GLN, Target Market ">

<title> Item Main Catalog </title>
```

*Table 1: DC Applicative example*

## 3.4 Conclusive Advantages and Disadvantages:

The benefits of using a systematic way of assigning and structuring metadata include:

- **Relevance:** Providing information that search engines can use to find relevant documents in large collections such as Web sites or document databases where text search alone brings up many irrelevant documents or lists of documents too long for users to look at.
- **Identity:** Providing descriptive information so that users can tell how old a document is, who wrote it, or how to get additional information. Most documents on Web sites now cannot tell the user whether they are 5 days old or 5 years old. Sometimes the user wants one, sometimes the other. Metadata helps a user know if the information is reliable and current.
- **Inventory:** A list of what information the Web holds so that the information can be managed, tracked, updated, analyzed and used efficiently.
- **Consistency:** The Dublin core, an international metadata standard, provides the framework and many of the rules for use so that metadata can be applied consistently in large and diverse organizations. This creates an environment in which users can search for and find information without needing to know which department produced it or to which program it relates.
- **Interoperability:** An international metadata standard such as the Dublin Core provides a way for information resources in electronic form to communicate their existence and their nature to other electronic applications (e.g. via HTML or XML) or search tools and to permit migration of information between applications or search systems.
- **Policy compliance:** A critical component of meeting for instance the Management of Government Information Holdings (MGIH) policy requirement to know and be able to find the information Governments hold.

One issue to be taken into consideration is the proper use of schemes, which are, as already mentioned, authorized lists of values from which content for metadata tags may be selected.

Because schemes are linked to their elements, misuse would have a negative impact on the reliability of search results and defeat the purpose of using metadata.

That said, the fact remains that this standard is limited by definition to the Internet resources, a mere piece of information which strengthens the hunt for a more global approach to adopt, one that could be conform to all sorts and origins of resources [3,4].

## 4. RDF APPROACH & XML IMPLEMENTATION:

RDF stands for Resource Description Framework Model, which was first accepted by W3C as a data model in February 1999. One way to proclaim its utility is by announcing that the next significant stage in the realization of the semantic web lies in extending the basic data schema representations with information on how the data can and should be used. This information can take various forms, including logical axioms, rules, constraints, or even functional and procedural representations.

### 4.1 Definition:

RDF is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web. RDF uses XML to exchange descriptions of Web resources but the resources being described can be of any type, including XML and non-XML resources

Descriptions used by these applications can be modeled as relationships among Web resources. The RDF data model defines a simple model for describing interrelationships among resources in terms of named properties and values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. As such, the RDF data model can therefore resemble an entity-relationship diagram. The RDF data model, however, provides no mechanisms for declaring these properties, nor does it provide any mechanisms for defining the relationships between these properties and other resources. That is the role of RDF Schema.

## 4.2 RDF Characteristics:

RDF is carefully designed to have the following characteristics:

**Independence**

Since a Property is a resource, any independent organization (or even person) can invent them. One called Author, and another called Director can be invented by different persons (they would probably only apply to resources that are associated with movies). Someone else can invent one called Restaurant-Category.

**Interchange**

Since RDF Statements can be converted into XML, they are easy for users to interchange.

**Scalability**

RDF statements are simple, three-part records (Resource, Property, value), so they are easy to handle and look things up by, even in large numbers. The Web is already big and getting bigger, thus there will probably be billions of these statements floating (literally). Due to all these facts, scalability is important.

**Properties are Resources**

Properties can have their own properties and can be found and manipulated like any other Resource. This is important because there are going to be lots of them; too many to look at one by one.

## 4.3 RDF Concepts Details:

Resource Description Framework, as its name implies, is a framework for describing and interchanging metadata. It is built on the following rules.

1. A **Resource** is anything that can have a URI; this includes all the Web's pages, as well as individual elements of an XML document. An example of a resource is a draft of any document at hand with its corresponding URL.

2. A **Property** is a Resource that has a name and can be used as a property, for example Author or Title. In many cases, all that matters is the name; but a Property needs to be a resource so that it can have its own properties.
3. A **Statement** consists of the combination of a Resource, a Property, and a value. These parts are known as the 'subject', 'predicate' and 'object' of a Statement. An example Statement is "The Author of http://www.textuality.com/RDF/Why.html is Tim Bray." The value can just be a string, for example "Tim Bray" in the previous example, or it can be another resource, for example "The Home-Page of http://www.textuality.com/RDF/Why.html is http://www.textuality.com."
4. There is a straightforward method for expressing these abstract Properties in XML, for example:

```
<rdf:Description about='http://www.textuality.com/RDF/Why-RDF.html'>
<Author>Tim Bray</Author>
<Home-Page rdf:resource='http://www.textuality.com' />
</rdf:Description>
```

## 4.4 RDF Classes and Properties:

An RDF Schema is expressed by the data model described in the RDF Model and Syntax specification. The schema description language is simply a set of resources and properties defined by the RDF Schema Specification and implicitly part of every RDF model using the RDF schema machinery.

The RDF Schema mechanism as a set of RDF resources (including classes and properties), and constraints on their relationships will be specified. The abstract RDF Schema core vocabulary can be used to make RDF statements defining and describing application-specific vocabularies.

The RDF Schema defined in this specification is a collection of RDF resources that can be used to describe properties of other RDF resources (including properties), which define application-specific RDF vocabularies. The core schema vocabulary is defined in a namespace informally called 'rdfs' in this example, and identified by the URI reference http://www.w3.org/2000/01/rdf-schema#. This specification also uses the prefix 'rdf' to refer to the core RDF namespace http://www.w3.org/1999/02/22-rdf-syntax-ns#.

As described in the RDF Model and Syntax specification, resources may be instances of one or more classes; this is indicated with the rdf:type property. Classes themselves are often organized in a hierarchical fashion.

Figure 2 below illustrates the concepts of class, subclass, and resource. A class is depicted by a rounded rectangle; a resource is depicted by a large dot. In the figure below, arrows are drawn from a resource to the class it defines. A subclass is shown by having a rounded rectangle (the subclass) completely enclosed by another (the superclass). If a resource is inside a class, then there exists either an explicit or implicit rdf:type property of that resource whose value is the resource defining the containing class. (These properties are shown as arcs in the directed labeled graph representation in figure 4). The RDF resources depicted in figure 3 are described either in the remainder of this specification, or in the RDF Model and Syntax specification.

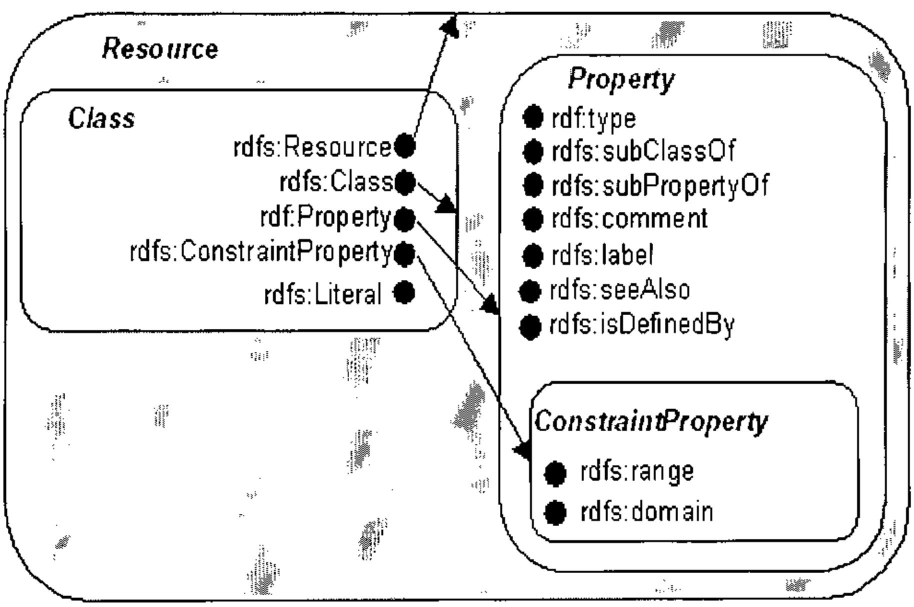


*Figure 2: Classes and Resources as Sets and Elements* [25]

Figure 3 shows the same information about the class hierarchy as in figure 3, but does so using a "nodes and arcs" graph representation of the RDF data model. If one class is a subset of another, then there is an rdfs:subClassOf arc from the node representing the first class to the node representing the second. Similarly, if a resource is an instance of a class, then there is an rdf:type arc from the resource to the node representing the class. Not all such arcs are shown. The arc to the most tightly encompassing class is shown, and the transitivity of the rdfs:subClassOf relation to provide the rest is relied on.

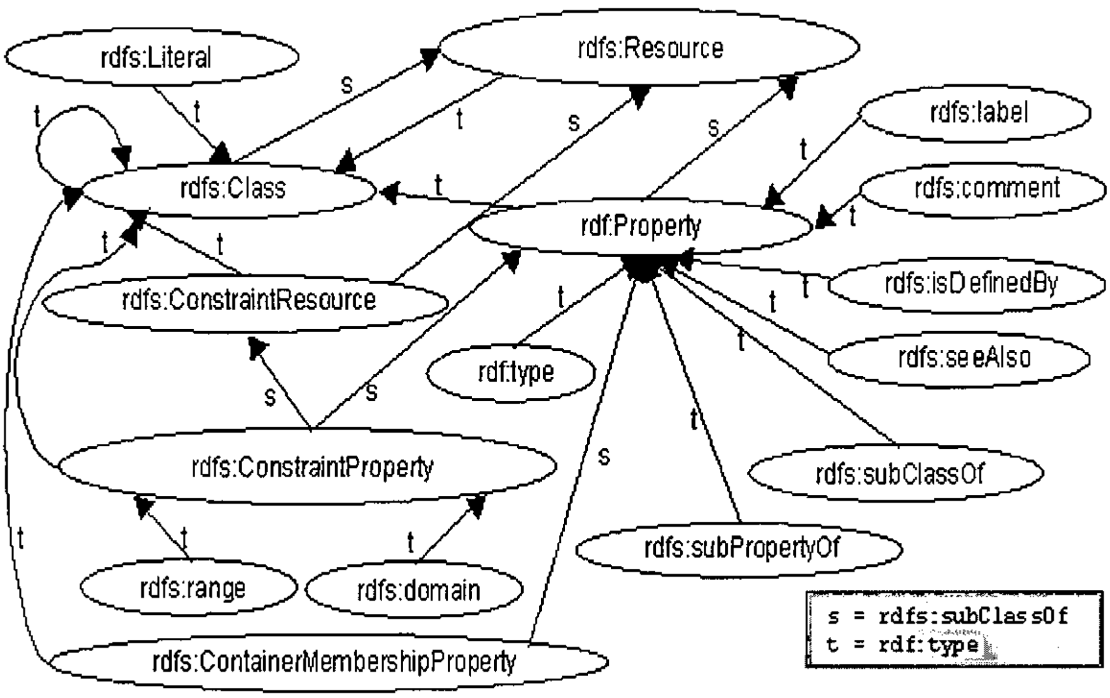


*Figure 3: Class Hierarchy for the RDF Schema*[6]

For instance, a subject taxonomy is a classification system that might be used by content creators or trusted third parties to organize or classify Web resources. The RDF Schema specification provides a mechanism for defining the vocabularies needed for such applications.

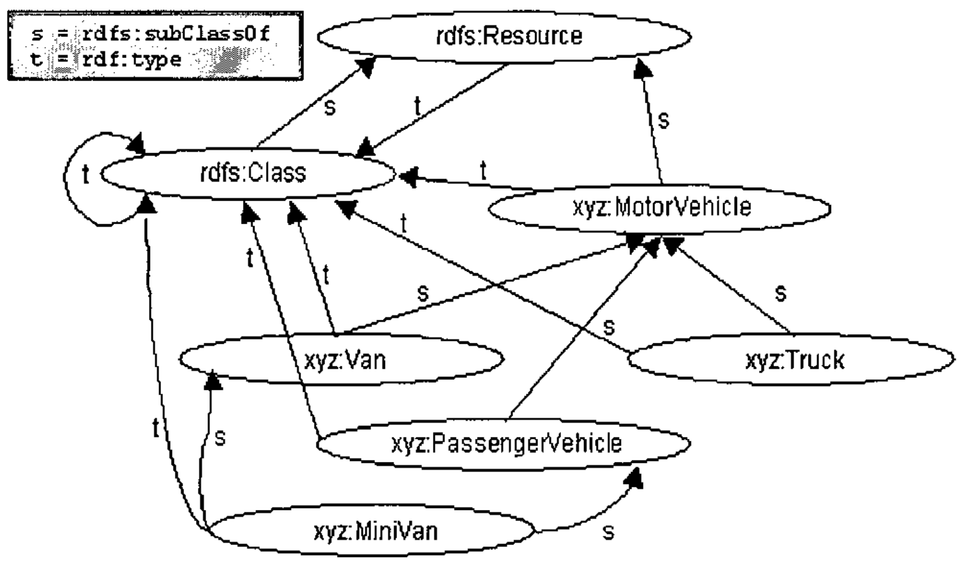


*Figure 4: rdf subClassOf* [26]

The RDF/XML shown here uses the basic RDF syntax of the Model and Syntax specification abbreviation mechanism provided by the RDF serialization syntax.

```
<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<!-- Note: this RDF schema would typically be used in RDF instance data
    by referencing it with an XML namespace declaration, for example
    xmlns:xyz="http://www.w3.org/2000/03/example/vehicles#". This allows
    us to use abbreviations such as xyz:MotorVehicle to refer
    unambiguously to the RDF class 'MotorVehicle'. -->

<rdf:Description ID="MotorVehicle">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 <rdfs:subClassOf
  rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Description>

<rdf:Description ID="PassengerVehicle">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="Truck">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="Van">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="MiniVan">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 <rdfs:subClassOf rdf:resource="#Van"/>
 <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdf:Description>

</rdf:RDF>
```

*Table 2: RDF/XML Syntax*

### 4.5 RDF Conclusions:

The RDF Schema Specification provides a machine-understandable system for defining schemas for descriptive vocabularies like the Dublin Core. It allows designers to specify classes of resource types and properties to convey descriptions of those classes, relationships between those properties and classes, and constraints on the allowed combinations of classes, properties, and values.

However, as is the case of the Dublin Core standard, RDF deals with Web resources, and sometimes systems that do not support XML or XML Schema requirements are available. This is why the search for a different approach is continued [6, 25].

## 5. OTHER KNOWN STANDARDS & APPROACHES:

Similar metadata standards exist. They are also known to provide a limited and subject-specific level of interoperability between systems and organizations to enable simple resource discovery. In what follows, some of these standards or areas are mentioned.

- METS: stands for Metadata Encoding & Transmission Standard. METS is intended to provide an XML-based language for encapsulating all descriptive, administrative, and structural metadata needed for the retrieval, display, management, and preservation of digital library objects. The Reference Model for an Open Archival Information System (ISO 14721:2002) defines the sum of metadata and data constituting a digital object as an "information package." It also delineates three major forms of an information package:

  1. A Submission Information Package (SIP), used to submit a digital object to a repository system

  2. An Archival Information Package (AIP), used to store a digital object within a repository.

3. A Dissemination Information Package (DIP), used to deliver a digital object to an end user.

METS was designed to fulfill the role of SIP, AIP, and DIP for digital library repository systems.

Other research initiatives are investigating extensions to METS to enable the preservation of audiovisual content or complex multimedia objects such as multimedia artworks. These approaches involve the association of ancillary and contextual information such as interviews with artists and the use of the Bit Stream Description Language (BSDL) to convert objects preserved as bit streams into formats that can be displayed on the current platforms [15].

- Rights Metadata: The Internet has been characterized as the largest threat to copyright since its inception. Copyrighted works on the Internet include news stories, software, novels, screenplays, graphics, pictures, usenet messages, and even e-mail. Rights metadata, like new emerging standards such as MPEG-21 and XrML, are specifically designed to enable automated copyright management and services [19].

- Multimedia Metadata: Audiovisual resources in the form of still pictures, graphics, 3D models, audio, speech, and video will play an increasingly pervasive role in every one's life and, because of the complex information-rich nature of such content, value-added services such as analysis, interpretation, and metadata creation become much more difficult, subjective, time consuming, and expensive. Audiovisual content requires some level of computational interpretation and processing in order to generate metadata of useful granularity efficiently. Standardized multimedia metadata representations that will allow some degree of machine interpretation will be necessary. The MPEG-7 and MPEG-21 standards have been developed to support such requirements[14].

- Metadata for Wireless Applications: Infrared detection and transmission can be used in libraries to beam context-sensitive data or applications to users' PDAs, depending on where they are physically located. Similarly, GPS information can be used to download location-relevant data to users' PDAs or laptops when they are traveling, for example scientists on field trips. Such context-sensitive applications require location metadata to be attached to information resources in databases connected to wireless networks. Extraction of metadata from real-time data flow, as well as high-speed metadata fusion across multiple data sensors are high-priority research goals within applications such as ROADNet [9].

## 6. CONCLUSIONS:

What has just been witnessed is a survey of representative metadata modeling efforts. The details of the most known approaches of metadata building and handling has been undertaken.

In the coming chapter, an enhanced global approach, which is without any doubt derived in some way from the existing ones (system and structured metadata repositories), comes to provide a more comprehensible clear step by step approach, with concrete data model, methods and examples to represent and implement it.

# CHAPTER IV

# METADATA MODELING

# PROPOSED APPROACH

# 1. INTRODUCTION:

This chapter introduces a proposed optimal metamodel based on a more global approach that can be appropriate to handle any kind of metadata, and satisfy the largest set of possible needs. Needless to say, as a first benefit, by adopting such a metamodel, a company will be able to expand its horizons, to increase its metadata applications, to switch between local, Web or any application metadata without having to develop and redevelop different metadata modeling standards and approaches to meet potential upcoming requirements.

The suggested metadata information model will be presented starting by introducing its step by step summarized algorithm, continuing its elaboration by deeply analyzing the methodology, presenting its conceptual model, as well as its corresponding data dictionary, and ending up with a straightforward applicative case study.

Advantages, disadvantages of the approach, along with a comparative study with the previous standards and modeling approaches will also be incorporated to magnify the proposal's approach.

In the next paragraph, the list of the main elements constituting the algorithm-like part of the methodology is stated, based on what will be called 'atomic elements'. Briefly, these elements are comprised by the conception of the most elementary business data and semantic terminologies, decomposed to their simplest atomic level of significance.

## 2. DESCRIPTION OF THE PROPOSED APPROACH'S STEP BY STEP ALGORITHM:

| |
|---|
| **a) Building the Elementary Data Taxonomy**<br>1. Conceiving the Atomic Elements<br>2. Uniquely Codifying Each Element<br>3. Assigning a Name for Each Element<br>4. Providing a Clear Definition for Each Element<br>5. Arranging Multi-Lingual Structures for These Elements |
| **b) Gathering Structure Data**<br>1. Identifying Meaningful Sets of Elements<br>2. Grouping Them into Entities (Tables or Views)<br>3. Assigning Them Meaningful Business Descriptions<br>4. Providing Multi-Lingual Facilities for These Terms |
| **c) Defining Data Stores**<br>1. Identifying The Most General Concepts of Business Perceptions<br>2. Uniquely Codifying These Large Groups<br>3. Assigning Convenient Business Terms<br>4. Providing Multi-Lingual Entries |
| **d) Liking Elementary Data (Taxonomy) to Structure Data and Data Stores**<br>1. Identifying Different Sets of Taxonomy Elements in Structure Data and Data Stores<br>2. Providing Different Sets of Synonyms and Taxonomy Definitions According to the Variety of Combinations<br>3. Pointing These Sets to URL Addresses |

**e) Putting together Business Areas**

1. Building the Largest Set of Useful Intelligent Business Areas
2. Grouping the Different Elements Codified into Different Angles
3. Inserting System Metadata Components (Physical Table Names, Column Names, Foreign Keys, Data Types, Etc…)
4. Pointing These Areas to Different URL Addresses for Web Metadata.

**f) Identifying Exclusions by Applications Clients and Web Users**

1. Identifying Different Applications Clients
2. Identifying Applications Users, Profiles and Privileges
3. Collecting Excluded Business Areas by Clients/Users
4. Collecting Excluded Structure Data by Clients/Users
5. Collecting Excluded Data Stores by Clients/Users

**g) Establishing Applications Processes and Web Resources Locations**

1. Identifying/Codifying Applications Processes and URLs
2. Linking These Identifiers to Structure Data
3. Linking These Identifiers to Taxonomy Elementary Data

**h) Creating Business Rules Semantic Extensions**

1. Identifying Business Rules Types or Categories
2. Identifying Themes and Contexts for Business Rules
3. Assigning Rules to Taxonomy Elements in Structure Data and Data Stores
4. Setting Business Rules Chaining and Inter-linkage
5. Arranging Structures and Entities for Different Rules Types Implementations
6. Documenting Business Rules

| |
|---|
| **i) Determining Information History Requirements**<br><br>1. Assigning Creators/Updaters to All Elements of the Whole Process<br>2. Assigning Creation and Update Dates to All Elements of the Whole Process<br>3. Providing Versioning Requirements and Necessities to All the Process |
| **j) Encrypting Metadata Main Elements**<br><br>1. Making Sure Metadata Critical Delicate Information Is Always Secure and Private<br>2. Applying Most Efficient Encryption/Decryption Algorithms in Order not to Allow Companies/Organizations Secrets' Exposure |

*Table 3: Approach's' Step by Step Algorithm Basic Elements*

# 3. ANALYSIS OF THE PROPOSED APPROACH:

## 3.1 General Introduction:

In the future, when metadata languages and engines are more developed, metadata should also form a strong basis for a web of machine understandable information about anything: about the people, things, concepts and ideas. This fact is kept in minds in the design, even though the first step is to make a system for information about information, inspired by Einstein's philosophy of simplicity.

For Einstein, as for many others, simplicity is the criterion that mainly steers theory choice in domains where experiment and observation no longer provide an unambiguous guide [22]. In other words, to divide and simplify elements to their most atomic level would be the unmistakable guide to the way for conquering any domain.

In the same context, the challenge throughout the whole effort is to make the complex simple. When new ideas and information are conveyed so that they are easily understood and used, the challenge has been met. The gathering of taxonomy and ontological elementary data constitutes the starting point; little by little the structure data is built, the different contexts, site maps and data stores, to finally get the process crowned with the bringing together of all available business rules.

## 3.2 Modeling Prerequisites:

This part aims at defining the prerequisites of the approach dealt with. They first address application metadata needs, then those of Web resources. Needless to say, the latter ones' prerequisites are less complex and demanding than the first mentioned, and that due to the limited end users' requirements, which can be summarized in Web documents search and statistical queries and reports.

In the case of application metadata, the data modeling methodology has an influence on the metadata collection phases. The gathering and analysis of info differs for example between a relational and an object oriented data model, but that does not imply that the metadata modeling approach will be affected.

However, the essential is to define a comprehensive model management strategy for conceptual, logical, and therefore physical models, to choose a modeling methodology, like for example Information Engineering or Enterprise Business Modeling, and get to decide on this modeling approach's naming conventions.

Similarly, for Web descriptive and semantic metadata, librarians ought to identify all their resources, set up naming conventions, authors' information, and have all kinds of statistics that are always useful for efficient metadata strategies prior or upon starting their implementation.

On the physical aspect, identifying the database engine's configuration parameters, the disks allocation and sizing parameters, all objects definitions that make up the physical layout of the database, its performance and stability, in short all kinds of usual DBA tasks, constitutes a contribution for the entire process.

## 3.3 Exploration of the Main Elements:

Having previously stated the major points that constitute the originality of the approach proposed, their exploration by defining, justifying and explaining the core elements that constitute their magnitude is next to be undertaken.

### a) Building the Taxonomy

This is the major most crucial triggering step that will affect all subsequent ones. Based on the strategy already explained, consisting of dividing and simplifying the elements to their most contextual atomic level, taxonomies consist of a structured vocabulary that identifies a single key term to represent a concept that could be described using several words.

For this purpose, a unique codification is provided for each taxonomy (elementary data). Synonyms and ontologies are assigned as names to identify all possible values of the taxonomy, all along with multilingual facilities and options.

As an important part of the semantic web, taxonomies entail adding an extra layer of infrastructure to the current HTML Web. These metadata vocabularies make it easier for databases to communicate with each other. A major problem with the Internet today is data fragmentation. With the Semantic Web, computers understand the meaning of a Web page by following hypertext links from Web documents to topic-specific ontologies. For instance, ontologies offer cross-references so a computer understands that "movie," "film," "flick," and "motion picture" are different expressions of the same concept.

**b) Gathering Structure Data**

In applications metadata more particularly, the step of identifying the structures in which the sets of taxonomy elements are grouped, in other words the tables or the views, is essential. Unsurprisingly, business and multilingual descriptions for these structure data should be provided, all in forms of classifications and access to information in all its various formats, types, manifestations, origins, and levels of completeness. Again, this step is important in the way that the vast flood of information that now permeates society enables anyone anywhere to obtain virtually any information at anytime.

**c) Defining Data Stores**

Data stores are large sets of structure and elementary data. They hold out the promise of extending nimble-fingered distinctions to large collections of data. Defining those helps a lot in the wide categorization of all kinds of elements, in order to increase the information integration capacity. Once again, multi-language business descriptions for these entities are assigned.

**d) Linking Elementary Data (Taxonomy) to Structure Data and Data Stores**

In this step, the elementary taxonomies and the structure data are merely being associated in their corresponding data stores and application menus. This is mostly used in applications metadata, where workflow needs necessitate such a step, in order to keep track of all the changes occurring at any level of the database, and be able to report any update, addition, deletion or cancellation of a given sensitive element, to the responsible person.

**e) Putting Together Business Areas**

So far, atomic elements have been merely described and spread in the metadata model. An open architecture that uses a business language that is incorporated within a multi-language taxonomy of business terms as well as a set of metadata which defines the elementary business data and links them throughout semantic layers.

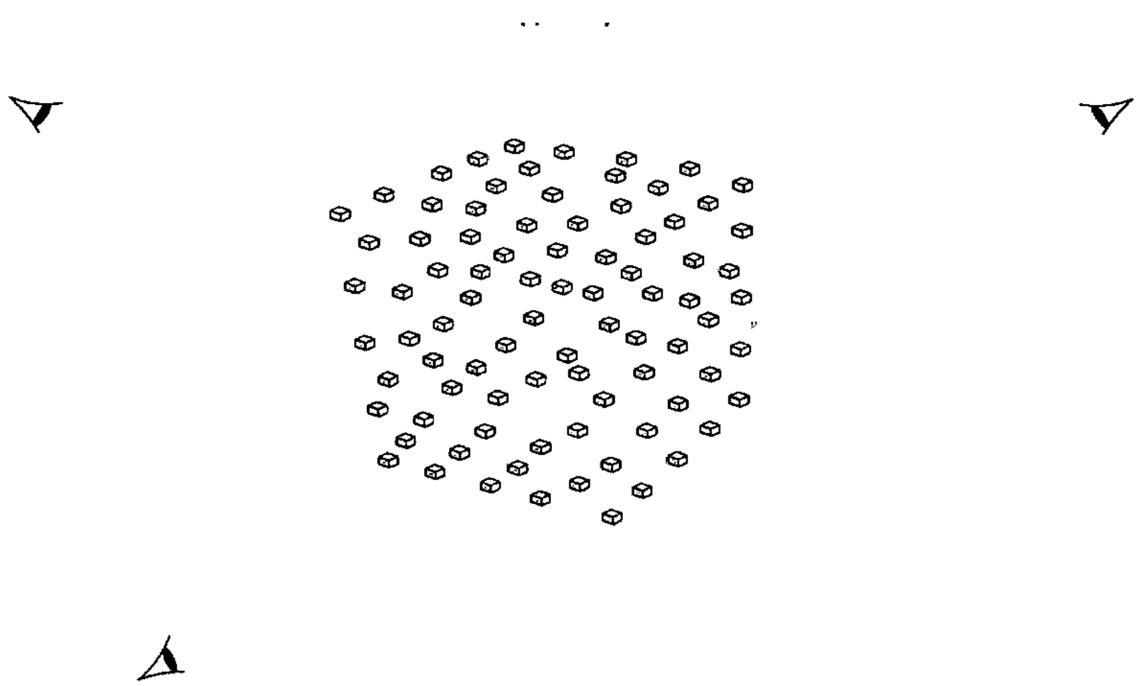

*Figure 5: Atomic Taxonomy Elements Flatly Spread*

The emphasis in this model is on standardizing the business terminology and then capturing rules.

The key here is that all these information will be intelligently grouped to be viewed according to different angles, whatever is the area of interest. As the fruit of experience, of dealing with users and clients needs, of responding to functional and performance commitments, the largest number of predefined business areas can be prepared by the metadata modeler in order to provide intelligent business information for decision support.

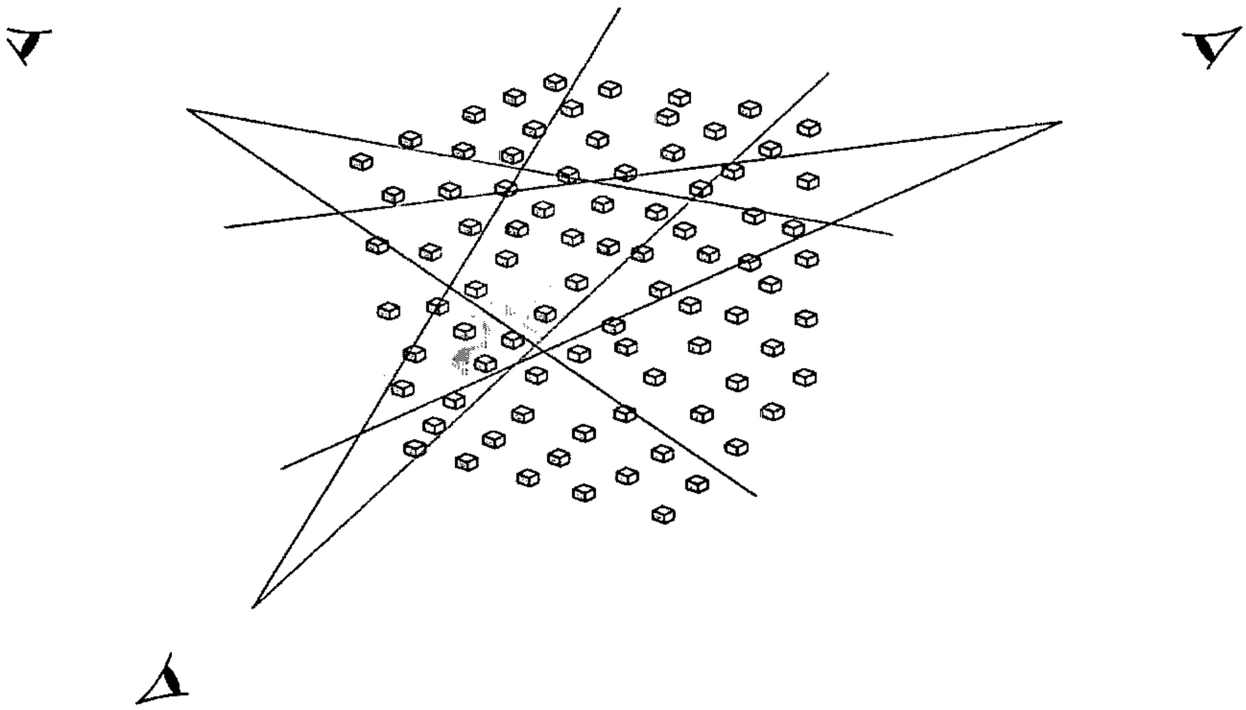

*Figure 6: Taxonomies Seen in Different Angles of Interest*

### f) Identifying Exclusions by Applications Clients and Web Users

Always in the context of different angles and views according to the user or clients needs, the ability to exclude business areas or rules specific for a certain user or client from the other ones is also provided, offering the user a complete vocabulary « home », and even better, making available some new grouping methods that help reorganize the information and get to navigate and know one's way around. This collaborates in fully satisfying users and clients' needs, and making it possible for this metadata model to be fully adaptable to their exact specifications.

### g) Establishing Applications Processes and Web Resources Locations

The locations at which the information described can be found constitute imperative elements throughout the metadata gathering process. Whether it is for Web or applications metadata - for search, reporting or querying purposes - defining and assigning applications menus or processes, servers' partitions and locations, URL and hypertext links, is a major step to be carefully and surely implemented.

### h) Creating Business Rules Semantic Extensions

Business rules are used to capture and implement precise business logic in processes, procedures, and systems (manual or automated). They can also provide the basis for expert systems.

Enterprises that take a model-based, architected approach to software component development can use business rules to refine the models and create better designs. An enterprise that properly documents its business rules can also manage change better than one that ignores its rules.

Business rules may be any of the following:

- Definitions of business terms
- Data integrity constraints
- Mathematical and functional derivations
- Logical inferences
- Processing sequences
- Relationships among facts about the business [1].

These types of business rules are examples of meta-data. They can be defined as meta-data, modeled as meta-data, and, most importantly, they can be implemented as meta-data for an enterprise's operational and strategic information management systems.

In general, in any organization, no central location is available to store all these business rules, and sometimes, it does not come as a surprise to find out that some business rules have been lost somehow!

Data analysts capture some business rules in data models and often document other business rules as text annotations. All in all, most business rules are coded into application programs and stored procedures - and lost! Now, when business operations are changed, it is the business rules that are mostly affected. At this very moment, one has to go business rule diving either in the programmers or analysts' heads, or in the sea of program code to find and change them.

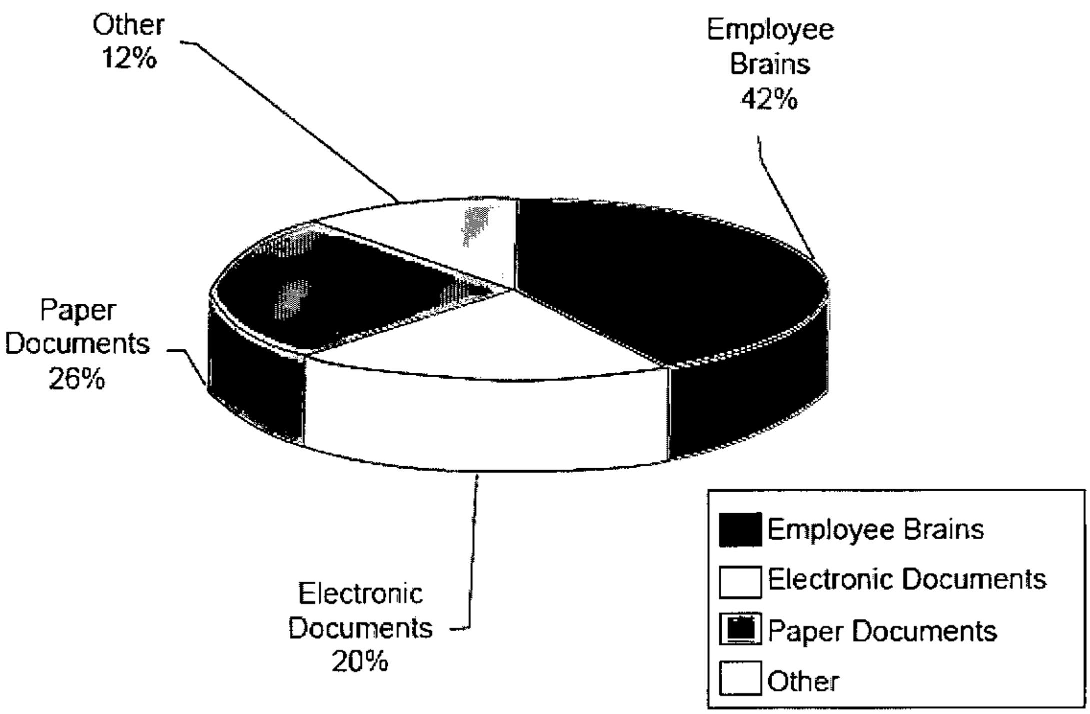


*Figure 7*: Metadata Distribution [27]

Implementing business rules as meta-data is the most rigorous and, at the same time, most flexible approach to business rule implementation.

The key to this is in fully understanding the rules, documenting them consistently, and building conceptual information models and logical data models that accurately reflect the rules. A critical success factor is in storing business rules in many formats in a single enterprise architecture model, being of course this metadata information model. This makes the rules *visible*. Naturally, rules are to be fully modeled, including specific values for meta-data. These values should then migrate automatically to any database design generated from the models and, ultimately, be automatically inserted into the database tables when they are created or altered from the design.

All these are factors that affect the design of the metadata and rules model, in identifying the rules contexts, roles, chaining and inter-linkage, in order to properly represent these rules and get the maximum gains out of this representation.

Moreover, the recommendations of the "Business Rules Manifesto" (expressed in the Appendix) are adopted and respected by the approach to stress and validate the its way of collecting and modeling business rules as metadata.

### i) Determining Information History Requirements

The ability to record each and every update and modification to the metadata and rules is essential all along the way of building the information metadata model. This step comes in parallel to all the previous ones, providing history entries that guarantee all changes are carefully kept and saved. One might refer to more advanced techniques, like the delta compression for example. But going into such details would be moving slightly out of the extents of this study.

### j) Encrypting Metadata Main Elements

It is a common widely known fact that metadata is very sensitive information. It entails business rules and data models that should not be revealed to the public world. For this reason, whenever an information model similar to the one described is being dealt with, the corresponding appropriate encryption and decryption algorithms should be anticipated, all throughout making sure all these confidential rules and information are securely kept private.

## 3.4 Analysis Conclusions:

This has been a way of introducing and going into the detailed features and aspects of what can be an optimal advantageous approach in the grouping, design and modeling of all this metadata, an approach that would provide the most accurate beneficial information to be used at the same time by applications developers and end-users, by web librarians and surfers, all taking into consideration previous standards and conclusions so far developed, whether in the system metadata or the approaches described...

Furthermore, the next sections of this chapter will contribute more in the illustration and general understanding of the whole proposed approach.

# 4. CONCEPTUAL PROPOSED MODEL OF THE APPROACH:

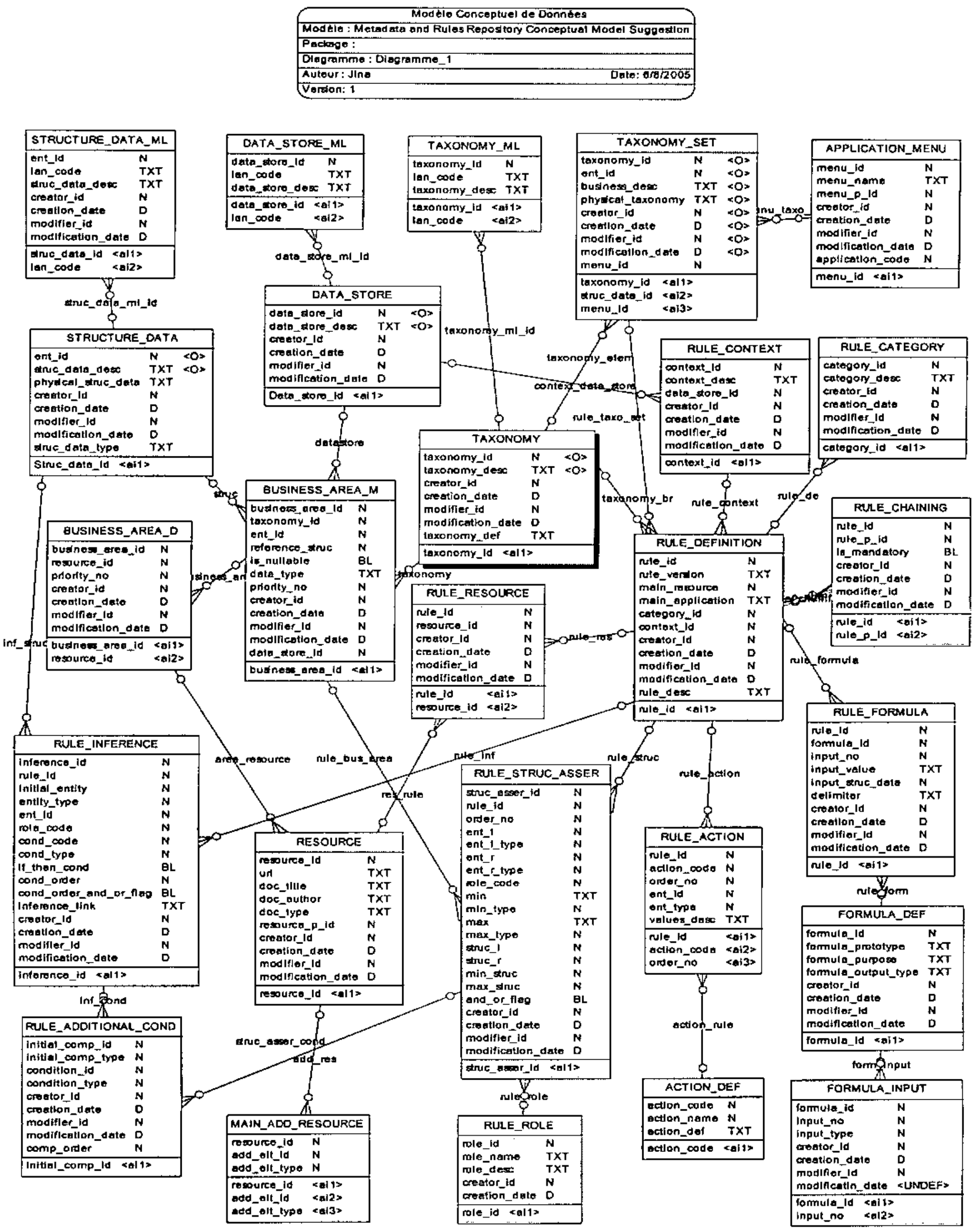


*Figure 8: Proposed Data Model of the Approach*

# 5. DATA DICTIONARY CORRESPONDING ELEMENTS:

In order to fulfill all the proposed approach's requirements and contributions, the model was meticulously conceived to incorporate system, application and Web metadata and rules.

Beginning with the main triggering step of all the remaining ones, which is the definition of all taxonomy atomic elements, entries and definitions, it continues with grouping this raw material into contexts and structure data, simultaneously associating them to their corresponding system data structures and resource links. It goes one step further to integrate the rules associated to all these taxonomy elements and business areas.

Needless to mention, the conception lies on a degree of respect of standard modeling methodologies and cohesive naming conventions.

## 5.1 General Notions:

Information History Entries:

In almost all structures, the presence of four main elements that keep track of all versioning and origins of all metadata and rules introduced:

- Creator_Id: Identification of the user who was at the origin of the creation of this data.
- Creation_date: Date at which this information was created.
- Modifier_Id: Identification of the user who modified.
- Modification_Date: Date at which the modification of the information occurred.

Multi-Language Structures:

Whenever business description entries are created, corresponding Multi-language parallel structures are being provided in order to guarantee translations of the different business terms and information in their right context.
A two-character ISO Language code is referred to in all such cases.

## 5.2 Main Structures:

### Taxonomy:

At the heart of the whole approach, the "Taxonomy" structure constitutes a complete codification of all the atomic elements.

The main related items are the following ones:

- Taxonomy_Id: An internal sequential identification of each atomic element.
- Taxonomy_Desc: A business English description of the element in question. Whenever standard designation is available (EAN.UCC, ISO, etc.), it is preferably used.
- Taxonomy_Def: A more detailed definition of the atomic elements.

### Application Menu:

This is the structure in which the menus of each application for which the metadata is designed are saved. Grouping and codifying each menu with its corresponding parent menu will allow by querying options to get tree structures of all menus and sub-menus. This is essential for Workflow activities and later linkage of business processes:

- Menu_Id: An internal code identifying a certain menu.
- Menu_Name: The name of the menu being defined.
- Menu_P_Code: The parent or higher menu immediately preceding the one in question.
- Application_Code: A pointer to the application to which this menu belongs. All applications that might be handled or dealt with in a listing structure have previously been identified.

### Resource:

This is a codification of all Web resources and links at which documents can be found. The idea is to codify these locations for easier later querying processing. The idea is inspired by the application menu's grouping and visualizing locations and links in a tree structure, connecting by prior accordingly to get any link's parents or sub-links. Moreover, several elements are simulated from the approaches described in the previous chapter (Dublin Core, etc.)

These resources are not specific for Web metadata. Nowadays, applications are often linked to e-learning HTML pages, or Help manuals. Moreover, if a certain company respects ISO standards or simply takes good care of all kinds of documentation, it will have this documentation carefully prepared and grouped in specific locations.

- Resource_Id: An internal code identifying each resource location. This is to simplify the access to this resource instead of writing the complete address link each time accessing it is needed.
- URL: the link or web address of the resource defined. If it does not start with www, then it is pointing out to an internal location on any hard disk or server at the company or organization's network.
- Doc_Title: The title of the page or document identified.
- Doc_Author: The author or originator of the page or document identified, in textual description.
- Doc_Type: The type of the resource. It could be an Excel file (.xls), a PDF (.pdf), an image, a Word document, an HTML page, etc. This way, all available types of documents will have been a priori identified and codified to be used in this location. Very often, in the case of application metadata, this type can point out to a DSS (Detailed Specific Study) containing all the details of the business rules and sections of a certain application.
- Resource_P_Id: The parent or higher-level resource immediately preceding this one. For instance, it may be the location of a document prerequisite to the one in question, containing previously described relevant business rules.

**Structure Data:**

This is where all structures or groupings of data are identified. This is related to the concept of table or view, with a business understandable description:

- Struc_Data_Id: An internal code identifying every structure data.
- Struc_Data_Desc: A description of the structure data.
- Physical_Struc_Data: The physical database name of the structure being described.

- Struc_Data_Type: This is the type of the structural data defined. For instance, it is "T" for Table, or "V" for View.

**Data Store:**

This is a high level identification of all concepts available and treated:

- Data_Store_Id: An internal code identifying a certain data store.
- Data_Store_Desc: A description of the data store.

**Business Area:**

A business area is a logical angle in which the business described is viewed.

According to users or clients specifications and needs, the grouping of metadata meticulously occurs. In this sense, elements of the taxonomy get to be in a certain normalized context, not necessarily reflecting the actual state of data objects.

The business area structures present elements found in a regular database metadata system, including data types, null property indicators, foreign key references and entities, etc.

The exact elements of this entity are not explicitly stated since they can be extensible according to the details degree needed upon implementation.

**Taxonomy Set:**

This is a composition of all sets available grouping different taxonomy elements in the various previously defined application menus. Taxonomy sets characterize exact representation of the grouping of the taxonomy elements as it is available in the database or data warehouse.

**History Elements:**

The sensitive nature of metadata information necessitates the creation of fields and elements that allow history management, updates tracking. In fact, delta compression techniques can be used for this purpose to get more accurate information. However, this study is limited to the use of four main elements by which history and updates are supervised:

- Creator_Id: Identification of the user responsible of the creation of this metadata.
- Creation_Date: Date at which this metadata was created.
- Modifier_Id: Identification of the user responsible of the creation of this metadata.
- Modification_Date: Date at which this metadata was created.

**Rules Categories:**

| CATEGORY_ID | CATEGORY_DESC | CATEGORY_TYPE |
|---|---|---|
| 1 | Definitions | Structural Assertions |
| 2 | Facts / Relationships | Structural Assertions |
| 3 | Processing Sequences | Structural Assertions |
| 4 | Formula / Calculations | Derivations |
| 5 | Inferences | Derivations |
| 6 | Integrity Checks | Action Assertions |
| 7 | Triggered or Timed Action | Action Assertions |

This categorization in the model is based on the fact that each Business Rule must be one of the following:

- A Structural Assertion — a defined concept or a statement of a fact that expresses some aspect of the structure of an enterprise. This encompasses both terms and the facts assembled from these terms.
- An Action Assertion — a statement of a constraint or condition that limits or controls the actions of the enterprise.
- A Derivation — a statement of knowledge that is derived from other knowledge in the business [2].

**Rules Variations:**

Their objective is to define a relationship between two given elementary or complex entities (rules, constants, etc.), and recursively: one identified relationship between 2 simple rules can itself be used in another more complex relationship being constructed. This relationship can have different roles, like 'Cardinality', 'Possession', etc.
Concrete examples will illustrate this idea in the case study provided.

| Rules Variations | | |
|---|---|---|
| **Min** | **Max** | **Rule** |
| 0 | 0 | Not allowed |
| 0 | 1 | Optional, may have one |
| 0 | X | Optional, may have up to X |
| 0 | N | Optional, may have more than one |
| 1 | 0 | Not meaningful (illogical) |
| 1 | 1 | Mandatory, one and only one |
| 1 | X | Mandatory, may have up to X |
| X | N | Mandatory, must have X or more |

To not that the previous ideas and categorizations are inspired by some business rules studies [1, 2]; they have been customized to meet the approach's needs.

## Input and Output Types:

Entities used in rules construction can be of several types. A given identifier can match different meanings according to its type.

Types can be something like the following:

| TYPE_ID | TYPE_NAME | TYPE_DESC |
|---|---|---|
| 1 | Constant | A certain constant |
| 2 | Percentage | A certain number identifying a percentage (%) |
| 3 | taxonomy_id | A Taxonomy element |
| 4 | struc_data_id | A Structure Data element |
| 5 | struc_asser_id | An already defined structural assertion |
| 6 | rule_id | An already defined business rule |
| 7 | Amount | A calculated amount |
| 8 | String | A certain string annotation |
| 10 | Role/Action | An action defined in the roles list, this action is to be undertaken when a condition points at it |

## Main Roles:

| ROLE_CODE | ROLE_NAME |
|---|---|
| 1 | Is |
| 2 | Has (Cardinality) |
| 3 | Results from |
| 4 | Belongs to |

Roles will be defined as needed. Their objective is to provide a semantic meaning to the different business rules and the relationships between them.

**Rule Contexts:**

Contexts are assigned to rules in order to group them into one large framework. Later on, a very useful user need might be to identify and extract all rules related to a certain context.

**Rule Actions:**

| ACTION_CODE | ACTION_NAME |
|---|---|
| 1 | Insert |
| 2 | Update |
| 3 | Delete |
| 4 | Commit |
| 5 | Validate |

**Rule Definition:**

This is the structure in which a rule is first defined and codified.

- Rule_Id: An internal code identifying every single rule.
- Rule_Version: The version corresponding to the rule. This is a concatenation of the rule's first internal code (when it was initially created), in addition to the version number, separated by a dash or any predefined delimiter.
- Main_Resource: The main resource in which this rule is described.
- Main_Application: The main application using this business rule.
- Rule_Desc: The business understandable definition most accurately describing the rule in question.
- Category_Id: The Category of the rule. A list of available rule categories will be provided subsequently. To note that the entity Rule_Category was specifically created for this purposes.

- Context_Id: The context, i.e. subject or area to which this rule belongs. A specific entity is also defined (Rule_Context) to group all possible contexts available and link them to their corresponding data store.

**Rule Chaining:**

In this entity, the chaining between the different rules is defined. Often in business, a certain rule comes as a fruit of another one, or causes a subsequent rule. This is why it is necessary to know the logical inter-relationship of rules:

- Rule_Id: The internal code from the Rule Definition Entity.
- Rule_P_Id: The parent rule or rules (a rule might have several independent rules that are prior to it).
- Is_Mandatory: A flag to denote if the first rule is mandatory to be fulfilled for the second to be valid.

**Rule Structural Assertion:**

Rules belonging to the structural assertion category are defined as follows:

| | |
|---|---|
| **struc_asser_id** | Internal identification code |
| **rule_id** | rule this structural assertion is defined for |
| **order_no** | order defined if same rule_id has several structural assertions |
| **ent_l** | left entity element id or code or number |
| **ent_l_type** | type of the left entity (could be constant, taxonomy, etc... from the above defined types) |
| **ent_r** | right entity element id or code or number |
| **ent_r_type** | type of the right entity (could be constant, taxonomy, etc... from the above defined types) |
| **role_code** | Role relating the right and left elements of this structural assertion |
| **min** | minimum variation between left and right |
| **max** | maximum variation between left and right |
| **struc_data_l** | struc_data_id in case the ent_l type is structure data |
| **struc_data_r** | struc_data_id in case the ent_r type is structure data |
| **and_or_flag** | 0=and, 1 = or, in case the same rule_id has several structural assertions |

## Rule_Action Assertions:

Rules of the category "Action Assertion" are defined as follows:

| | |
|---|---|
| **action_asser_id** | Internal Identification Code |
| **rule_id** | The rule of category relating to Action Assertions |
| **action_id** | action indicator (already defined) |
| **order_no** | order to proceed in case several actions relate to the same rule |
| **Ent_id** | The element (most probably table) the action will affect (especially in the case of insert, update, etc.) |
| **Ent_type** | To indicate what sort of entity is being affected (table or struc data, another rule, etc...) |
| **Values_desc** | Set of values needed for insert or update -- this can be further developed and structurally decomposed. |

## Formula Structures:

Formulas have their own appropriate way of definition due to the specificity of the elements and parameters that constitute them.

Formula Definition:

| | |
|---|---|
| **formula_id** | Internal Code for formula identificaton |
| **formula_prototype** | Prototype of the formula |
| **formula_purpose** | Purpose description |
| **formula_output_type** | Output Type (From the list of types defined) |

Formula Input:

| | |
|---|---|
| **formula_id** | Formula Identification |
| **input_no** | Input number by formula |
| **input_type** | Input type (from the list of types defined) |
| **Delimiter** | Delimiter operator separating the inputs |

## Rule Formula:

Assigning a rule to its corresponding formula is the matching described in the below structure.

| | |
|---|---|
| **rule_id** | Rule Identification |
| **formula_id** | Formula Identification |
| **input_no** | Input Identification |
| **input_value** | Value of the input in question |
| **input_struc_data** | Table or structure data in case the input type defined during the formula input identification is "struc_data" |

## Rule Inference:

Rules that are something like 'IF/THEN/ElSE' are defined below, with a possibility of recursively calling the conditions of different types as needed.

| | |
|---|---|
| **inference_id** | Inference internal |
| **rule_id** | Rule Identification |
| **initial_entity** | Entity conditioned in the IF Part |
| **entity_type** | Type of this entity (from the type list defined) |
| **role_code** | Role Identification from the list defined |
| **cond_code** | Entity conditioned in the IF Part |
| **cond_type** | Type of this condition (from the type list defined) |
| **if_then_cond** | Boolean indicating if further conditions are appended to this rule |
| **cond_order** | Order of the different conditions appended to this rule's inferences |
| **cond_order_and_or_flag** | Indicator whether the linkage between the different conditions is "or"/"and" |

## 5.3 Conclusions:

In conclusion, it is essential to mention that the data model described in the previous section is an extract sample one that could surely be further extended and developed to meet the proposed methodology's details. For instance, one matrix structure encompassing the common properties of each element will sure meet the requirements of a large organization with a multiplicity of needs and applications. In a word, the main purpose of providing this data model and its corresponding data dictionary is to prove the possibility of the approach's implementation and by this make it appear more concrete and familiar to the reader.

# 6. A STRAIGHTFORWARD CASE STUDY:

One familiar example to illustrate the approach proposed consists of extracts or instances of any traditional university system.

In order to demonstrate it, the considered case will briefly be analyzed, implemented in order to provide the corresponding physical model and data dictionary, the major related business rules will be defined next, to finally get the step by step approach followed. Thus the objective is to clarify as much as possible the way metadata information modeling is suggested to be optimally taking place.

## 6.1 Brief Analysis:

At the heart of any typical university system, the major entities that define it consist of courses, students, faculty members, majors, classes, semesters and time sheets, all cooperating together and defined in the coming few lines.

Any student should have the main information that identify him, including Id, name, personal information, major, classes he registers in during a given semester, corresponding grades, all in all well defined and systematically saved.
In parallel, the same handling should be foreseen for faculty members, and in particular instructors, who should be identified, along with the fields they have specialized in, the courses they teach, etc.

On the other hand, a list of all majors offered by the university, along with the number of credits constituting them, the departments they relate to, should be prepared.
Similarly, the courses should be identified and classified according to their nature, and the number of credits composing them.

Moreover, the association of a major with a course provides a different type of this course. For instance, an Engineering course is classified as a major course for the Civil Engineering major, while it is obviously an elective course for the major of Computer Science.

During a given semester, several courses are opened, to be given by appropriate instructors, in a scheduled time. Several sections of each class can take place in different classrooms. All this registration information will be associated together, in order to be able at the end to assign the corresponding grade for a student to the different courses he is taking per semester.

Needless to say, in this example, the study will be limited to regular typically occurring cases, the university system by itself being of course out of the scope of the whole study.

## 6.2 Implemented Physical Data Model:

Correspondingly to what has just been described, the following physical data model consists of an implementation of all implied structures, related to students, faculty members, courses, majors, semesters and the different main interrelationships between the various entities.

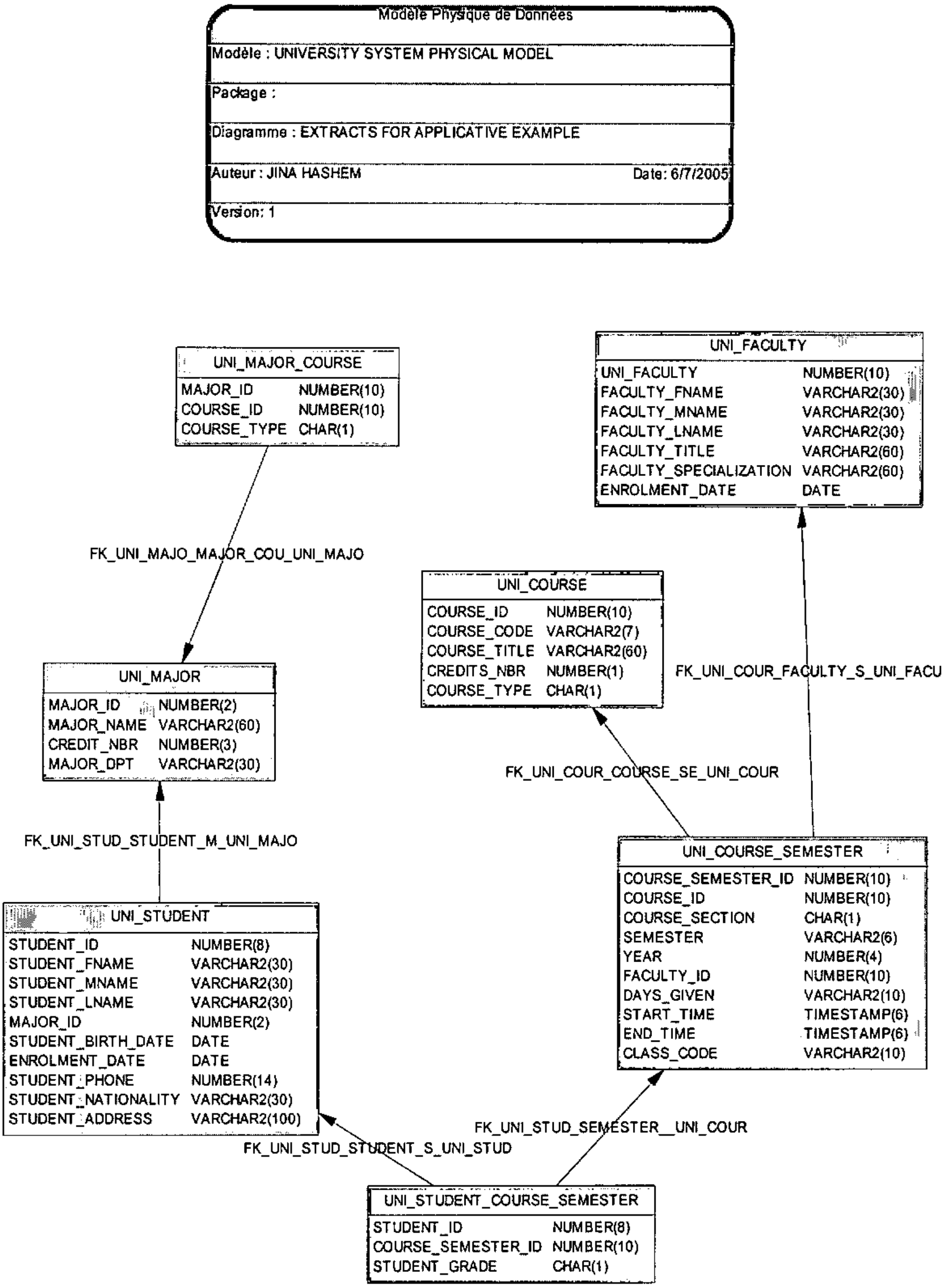


*Figure 9:* Physical Data Model Illustration

## 6.3 Appended Business Rules:

To go one step further, the main business rules linked to all these elements and entities are identified.

- A student, during a given semester, can take at least 9 credits, and at most 16.
- The total number of students in a given class should not exceed 35.
- The cumulative average of a student is gotten by calculating the sum of grades gotten on all courses taken, divided by the overall number of credits (supposedly).
- A full-time instructor is not allowed to give more than 4 courses per semester.
- The letter grades have specific conversions (A:4, B:3, C:2, and D:1)
- If teachers are late for more than 15 minutes, students are allowed to leave.
- According to its course's type, a credit has various fees amounts. (An engineering course costs $250; a regular course costs $200, etc...)
- F is always a failing grade
- In case the course is a major one, then the failing grade is C.
- A student who misses 6 classes has to drop the course.
- A GPA greater than 3.5 implies a 50% discount on tuition fees.

## 6.4 Approach's Step by Step Application:

Having identified the raw material necessary to start conceiving and modeling the information metadata and rules, the step-by-step implementation is started, according to the proposed detailed method or approach.

It is important to mention that the example adopted is a typical application metadata one, where usually the physical model would have been most probably previously conceived. Knowing that, it is crucial to mention that sometimes information metadata can be managed without returning to the physical data model, but directly after or during the analysis phase, where the opportunity of immediately identifying taxonomy and business elements can positively constitute a very advantageous shortcut.

- **Building the Taxonomy:**

  This is the step that consists of conceiving all atomic elements needed or deduced from the analysis and model overview.

  One thing that is worth mentioning at this stage is the fact that taxonomy elements to be conceived definitely do not have to exist in the physical data model. They can be logically deduced, they could even sometimes be calculated from another taxonomy set. This is an extra proof showing how much system metadata has missing elements and factors that prevent from proceeding with the information warehouse aimed for.
  In the example, the following instances are collected, codified, described, translated and defined:

| Taxonomy Elements | | |
|---|---|---|
| 100 | Student Id | Unique Student Identification Number |
| 101 | Student First Name | First Name of the Student |
| 102 | Student Phone | Student's Phone Number |
| 103 | Student Address | Student's Textual Address |
| 104 | Student Nationality | Student's Nationality |
| 105 | Faculty Id | Unique Faculty Identification |
| 106 | Faculty Name | Name of the Faculty |
| 107 | Passing Grade | Passing Grade |
| 108 | Failing Grade | Failing Grade |
| 109 | Credit | Credit Element to be Taken |
| 110 | Cost of a Credit | Credit's Cost Amount |
| 111 | Cumulative Average | Cumulative Average over all the semesters a student was registered in |
| 112 | Semester | Semester Identification |
| 113 | Course Type | Type of the course (Masters, Major, Elective) |
| 114 | Start Time | Time at which a class starts |
| 115 | End Time | Time at which a class ends |
| 116 | Year | Year Number |
| 117 | Major Id | Unique Major Identification |
| 118 | Major Name | Descriptive Name of the Major |
| 119 | Major Credits Number | Number of Credits that Form a Major |
| 120 | Course Id | Unique Course Identification |
| 121 | Course Code | Textual Unique Descriptive Code of the Course |
| 122 | Course Type | Class taking place at a certain time, in which students are registered |
| 123 | Department | Department Descriptive Identification |
| 124 | Faculty Specialization | Specialization of the faculty member (Mathematics, Computer Science, etc.) |
| 125 | Faculty Enrolment Date | Date at which a faculty member joined the university |
| 126 | Student Enrolment Date | Date at which a student was enrolled |
| 127 | Course Section | Section of a Given Course During a Semester (A, B, etc...) |
| 128 | Faculty Position | Position of the Faculty Member (Associate Professor, Assistant Professor, etc..) |
| 131 | Student Middle Name | Middle Name of the Student |
| 132 | Student Last Name | Last Name of the Student |
| 129 | Student Grade | Grade Taken in a Certain Class |
| 130 | Student Birth Date | Birth Date of the Student |

*Table 4: Taxonomy List - Example*

- **Gathering Structure Data:**

In this example, all deduced structure data can be defined as tables grouping the related elements of the same nature. Thus, similarly, all entities are identified, translated and defined:

| **Structure Data** | | |
|---|---|---|
| 20 | Courses Main Info | UNI_COURSE |
| 21 | Courses Identification by Semester | UNI_COURSE_SEMESTER |
| 22 | Faculty Main Info | UNI_FACULTY |
| 23 | Majors Identification | UNI_MAJOR |
| 24 | Majors' Courses | UNI_MAJOR_COURSE |
| 25 | Students Main Info | UNI_STUDENT |
| 26 | Students Courses Details by Semester | UNI_STUDENT_COURSE_SEMESTER |

*Table 5: Structure Data List - Example*

- **Defining Data Stores:**

The main data stores identified during the process of metadata gathering and modeling of this example reside in:

1. Student
2. Faculty
3. Major
4. Course

For a short while, in the case of this example, one might be confused between the concept of data store and the one of structure data; but that is only due to the fact that the case elaborated is a restricted one. In fact, the structure data that might generate some confusion are the main central ones belonging to their corresponding data stores. Typically, these are the ones on which triggers will be placed to help Workflow applications identify any majors operations occurring in every data store. For example, upon the detection of a creation or deletion (cancellation) of a certain course or major, a notification message will be prepared and scheduled to be sent to all concerned faculty members...

- **Linking Elementary Data (Taxonomy) to Structure data and data stores:** In a very straightforward step reminding of a database server's system dictionaries, but more developed to fit to the semantic and business nature of the information, the taxonomy elements are linked to the different tables they are physically related to, while the corresponding exact physical name related to them is provided.
For instance, here is a representation of the student structure data being associated to its taxonomy elements:

| Taxonomy_Id | Ent_Id | Physical_taxonomy | Menu_Id |
|---|---|---|---|
| 100 | 25 | student_id | 1234 |
| 101 | 25 | student_fname | 1234 |
| 131 | 25 | student_mname | 1234 |
| 132 | 25 | student_lname | 1234 |
| 102 | 25 | student_phone | 1234 |
| 103 | 25 | student_address | 1234 |
| 104 | 25 | student_nationality | 1234 |
| 117 | 25 | major_id | 1234 |
| 126 | 25 | enrolment_date | 1234 |
| 130 | 25 | student_birth_date | 1234 |

*Table 6: Linking Elementary and Structure Data*

To note that virtual Menu Identifications are assigned since the application is not available.

- **Putting together Business Areas:**
One interesting business area or angle or view for this information is a list or a report of all courses given by an instructor in a given semester. Thus, this triggers to create a business area containing the following set of taxonomies:
  1. Faculty Identification
  2. Faculty Name (and any other additional required information)
  3. Semester Name

4. Course Identification
5. Course Name

This set of elements will be given a defined group code to illustrate their belongings to the same business area.

On the other hand, another business area may be to get a list of all students or instructors enrolled or giving courses during a semester. This would imply the creation of a business area that includes:

1. Students or Faculty Identification Taxonomy Elements
2. Faculty Name (and any other additional required information)
3. Semester Name

Interestingly enough, the maximum number of predefined logical business areas would be beneficial to build, merely to facilitate the user's job in the future, minimize processing, and have all the information that could serve his needs ready for use.

- **Identifying Exclusions by applications clients and Web users:**

This step is not very applicative in the restricted case study, because the entities being dealt with are usually common for all universities.

However, in case a much complete system where some entities or elements are specific for one university or department and not applicable to another, then structures to identify the different universities on one hand, and on the other to point at the entities and elements to be excluded for each university's needs, will have to be locally prepared when applicable.

Now, if the exclusions for Web users are needed to be specified, for example faculty members or students accessing the metadata for reporting or querying purposes, the access on the metadata, especially according to each university's policy might need to be considered. For instance, it is very probable that students might not be given access to view the grades of other students or classes. To fulfill this need, an exclusion by user structure will include the student identification along with the grade taxonomy.

- **Establishing applications processes and Web resources locations:**

  In an advanced stage of the analysis, having designed the screens, the hierarchy that links them, and on the other hand in case the application is needed to be placed on the Web and have all its corresponding documentation presented, along with e-learning tools and files, placing them on specific locations of the servers should be planned. All these locations, screens hierarchy identification, should be codified and linked with the taxonomy elements, documents and resources in order to automatically later access any field or subject needed.

- **Creating Business Rules Semantic Extensions:**

  Upon collecting all relevant business rules, it is major to assign them with their corresponding type and definition.

  Contexts for available business rules can for instance be like one of the following:

  - Student Registration
  - Graduation Rules
  - Etc.

  In this example, here is the set of business rules identified and this is the course of actions to be done in order to insert them in the repository according to the modeling approach proposed:

  *1. A student, during a given semester can take at least 9 credits, and at most 16:*

  This is a rule belonging to the category 2 (i.e. Facts / Relationships). Its Rule_Id is 3, its version is 3.1 (Rule 3 concatenated with 1 implies at the first version of the rule 3), supposing it is the third rule being defined.

  This rule is stated in the Resource_Id 30, where the link to the university's Rulebook was stated.

  The rule's description can be something like what has just been stated above in italic characters.

  Next stage is to implement the exact details of the rule.

Two main taxonomies can be identified: Student Id, and Credit, with a relationship between them having a minimum value of 9 and a maximum value of 16, with the Role: Can take, already defined with the ID of 5.

*2. The total number of students in a given class should not exceed 35, or be below 12.*

This is another rule of the category 2: "Facts/Relationships", but based on another sub-rule, of the type function or calculation, where the main elements are Count(Student_Id) and Class, with a minimum and maximum of 12 and 35 respectively.

First, the rule function "Count(Student_ID)" is defined with an ID and a prototype, then its input of type taxonomy is assigned, next it is linked to the rule created by assigning the taxonomy student ID to the unique input needed, with the delimiter pointing at the function Count().

Going back to the main rule, the role linking the min and max is "1=Is".

*3. If Grade = F then student fails the course*

This is a rule of type "Inference", it is a combination of two facts/relationships where the occurrence of the first implies the second.

The first fact is: Grade (which is a taxonomy element) - is equal (the role with id = 1) - F (taken as a constant type).

The second relationship is: Student (taxonomy) - fails (role to be defined) - Course (the second taxonomy).

In the "inference" structure, while the first fact is the condition code with type = structural assertion, the second is the initial entity with same type, resulting from the fulfillment of the previous condition.

*4. If GPA is between 3.5 and 3.6, then discount = 50%*

Again, this is an inference rule composed of two facts/relationships. In fact, the result fact is that the discount taxonomy is equal to 50 (of type percentage).

As for the initial condition, it is the GPA taxonomy being between two constant minimums and maximums, which are 3.5 and 3.8.

Similar rules can be deduced from various values of GPA, which calculation is in itself a rule specifically defined.

Therefore, a rule chaining between the GPA calculation and this rule implemented can be made, where the first one is a mandatory parent rule to be identified before the latter one.

- **Determining information history requirements:**

  Practically visible in the four repetitive fields of the conceptual model (pointing at creation and modification dates and achievers), keeping record of metadata generators and updaters help in tracking any addition, change, or perhaps erroneous information. For example, a certain university rule is created by a known user who will be responsible for this rule's creation. Same goes for an alteration of a rule, a certain user known by the system will be held accountable for this modification which will by nature have an effect on all university staff and students.

- **Encrypting metadata main elements:**

  As a general rule, it is mainly imperative to always be aware about the sensitivity, the value and the magnitude of the handled metadata. For this reason, well-considered encryption and decryption algorithms to keep Students, Professors, University Rules' data secure should be anticipated. The technical details related to this issue do not constitute an objective to be tackled throughout this thesis; the essential is to manage the right algorithm in order not to be revealing private info, or allowing potential attacks.

# 7. APPROACH'S VALIDATION, ADVANTAGES, DISADVANTAGES AND COMPARATIVE STUDY:

In this section of the chapter, validation elements, as well as the main disadvantages and advantages of the proposed approach will be stated. A comparative study of the offerings of the discussed approach relatively to the previous ones already discussed will then be conducted.

As stated several times earlier, the approach tends to meet the requirements of the largest possible set of needs expected from a metadata model, not to mention the business rules extension, which by itself can be considered as an achievement.

## 7.1 Emphasis on the Approach's Validation:

The case study, descriptions and arguments presented aimed at validating the approach's methodology. On the other hand, validation of the metadata itself is necessary when it is transformed into its semantic format. This is known to be a difficult and laborious task given the large and complex elements, taxonomies, rules handled and databases in use. Specific routines detecting any possible errors in the metadata, techniques identifying terms and relations, physical inconsistencies can be developed, but to a certain extent, for when it comes to the semantic meanings, to the business rules validity, the call usually goes for an expert or somebody known as a cognitician or a lexicographer. A cognitician is someone who is professional in cognitive sciences, with very particular delicate tasks in which scientific competences are fundamental.

## 7.2 Proposed Approach's Disadvantages:

### Cost Factor:

The fact that the approach can accommodate a large set of requirements has required the addition of several elements and units to the modeling approach. This richness necessitates on the other hand extra work and effort in order to fill and provide all related metadata elements that play a role in the completion of the model. This, without any doubt, contributes in the increase of the cost factor related to the whole process of metadata gathering and modeling.

**Originality Factor:**

The proposed approach, although in some aspects derived and inspired by the elements of other already discussed approaches and standards, can still be considered original and fresh new relatively to those. Therefore, organizations - having already gotten accustomed to existing approaches - may find it hard to adapt to a new approach, even when the benefits of the new one outdo by far those of the former ones.

**Quality Factor:**

Just like in any metadata system, assuring an optimal quality is an issue to be seriously and constantly considered. It is true that in the approach some predictive measures to control and report metadata inconsistencies or redundancies can be taken; even though, lack of quality may still occur; missing or incomplete entries, with all kinds of heterogeneities involved and semantically related, if not persistently checked and verified, might turn out to be unavoidable.

## 7.3 Proposed Approach's Advantages:

First of all, it is crucial to note that the cost of the proposed approach can be reduced according to the organization's needs and policies. For instance, it is up to the organization that will apply and use the metamodel to decide to which extent and how much deep it will go in its implementation. This taken into consideration, cost can be critically reduced. In addition, it should be known by now that cost should not be measured independently, but rather the cost of implementing the system should be compared to not implementing it, or to implementing it versus implementing other tools and standards available in the market, known to be very expensive (examples of tools are JLOG Rules, Embarcadero System Metadata, etc...)

On the other hand, concerning the quality of the metadata, the atomicity of elements - in other words the key concept that triggers the approach's remaining principles - contributes in reducing the redundancies or inconsistencies, not to re-mention the different routines and procedures that limit quality decrease or problems.

Another crucial point related to the originality of the approach relies in its being more global and in a way derived and encompassing the other existing ones. That said, and apart from the above, as well as the previously mentioned advantages offered by the metadata concept by itself, the light can be shed on the following aspects that are put forward by the proposed approach:

**a) Interoperability and Communication with Other Systems:**

Uniquely defining the atomic elements that may be present in a variety of sources and systems is the key factor that makes it possible for systems to communicate the existence and characteristics of information resources to other applications or search tools. In the same context, the approach also permits efficient migration of information from one application or search system to another. The ability of systems to communicate with other applications, search tools, systems, etc., is known as interoperability.

**b) Traceability, Flexibility and Integration:**

Global competition and changing business scenarios are forcing companies to reevaluate and change the methods they use to compete. As a result, business is demanding more functionality and shorter development life cycles from their IT departments to attain and maintain competitive advantage in the marketplace. These demands have forced the IT departments to have a closer look at their organization's information systems.

The proposed metadata approach's repository significantly reduces both time and cost of analysis and development by documenting the data transformation rules, data sources, data structures, and context of the data in the Web, in data warehouses and data marts. This information is critical because without complete and atomic metadata, the transformation rules often reside only either hard coded or in the staff's collective memory. Because the results of the analysis and developmental changes are captured and retained in the metadata repository, the benefits are long lasting, helping to reduce the costs of future releases and the likelihood of developmental errors.

**c) Ability to Generate Existing Common Standards Elements:**

Being perfectly aware that adopting a metadata standard ensures that metadata in different systems and Web sites will be coordinated and client-focused, the approach is designed in a way to allow the generation of the different elements known in a common standard. Specific algorithms for each standard can be design to allow the extraction of the Dublin Core elements for example. This again is a proof of the hypothesis already stated proposing that the approach recommended addresses the different subjects in a more global way.

**d) Better Business Value Delivery and Employee Turn Over Impact Reduction:**

One of the major challenges that the businesses face today is the high rate of employee turn over in IT departments. When a company loses an employee, it is losing much more than an individual who has programmed some Java or any other language code for the past three years. In reality, it is losing an individual who has accumulated three year's worth knowledge about the company business and its business systems. This metadata approach seeks to capture the knowledge – both business and technical – that is held with the employees, and make it accessible to everyone.

When looking at where knowledge exists within companies, it is clear that the vast majority of knowledge is stored within the minds of the employees. On the other hand, the risk of not getting the information correctly or accurately delivered by employees is always there.

In conclusion, capturing the knowledge from the minds of the people is a challenge in the proposed metadata approach to fulfill the mission of providing better business value and reducing the negative impact of employee turn over.

**e) Didactic Property:**

The development of metadata schemas has been at the core of the learning technology standardization process [23].

In the context of the proposal, the light is shed on the instructive or didactic property of the approach itself, in terms that it was presented in a step by step way to make a global in a way tutorial-like study introducing the interested reader to the whole concept of metadata, its design and gathering process, not to mention the part specific for business rules and their categorization, implementation, chaining, etc...

**f) Versioning, Growth, Evolution and Change Management:**

Decision support systems and their underlying data warehouses are growing very quickly and in directions that are difficult to predict, which makes the IT department's task of maintaining and extending them very difficult. This trend is likely to continue, as corporations become more customer focused and demand a better understanding of their competitors and their customer base in order to reach sound strategic decisions.

The approach proposed provides corporation's IT department with the ability to maintain and grow its system over time. It allows corporations a much greater level of knowledge about their decision support architecture. This information enables companies to make better architectural decisions and add greater flexibility to change the decision support systems.

Moreover, it is commonly known that by capturing their business rules as meta-data, enterprises consistently deliver effective strategic information -- on demand, to the right people, in the right format. They also control the only constant in enterprise management: Change. Rules are implemented in this metadata model, therefore support for business analysts when planning changes is granted.

**g) Conformity and Adaptability with Grid Metadata Requirements:**

In this study, the quest is not to elaborate the specifications of the grid conception. However, based on the Proceedings given in the 2004 SSDBM (more particularly the 16$^{th}$ International Conference of Scientific and Statistical Database Management), the way the approach proposed suits the requirements of the Grid Metadata management will next be shown.

There are several requirements for managing metadata on the Grid. They can be broadly grouped in four main categories:

- The need to store and share the metadata.
- The need to organize the metadata in a logical fashion for ease of publication and discovery.
- The need to customize the view of the data by individuals.
- The need to support metadata about large-scale data sets [21].

That said, the fact that this approach, by being at the same time centralized, atomic and global, satisfies the requirements cited above can be shown. In fact, the centralized repository makes the storing and sharing of metadata possible; the different business areas grouped structure data, data stores give the modeled metadata a logical easy to publish and discover organization. Moreover, the already elaborated different views and angles of taxonomies and business areas provide the needed customized individual information view, as well as the large-scale data sets support.

## 7.4 Comparative Study:

In order to fully understand and justify the value that the proposed metadata solution gives, it is necessary to have a way of quantifying the results. One way to do that would be to compare the proposed solution to other solutions or categories of solutions available and already overviewed. The below table provides this comparison.

| Feature/ Attribute | Proposed Approach | System/ Application Metadata Tools | Web Metadata Tools & Standards | Relative Details & Justifications |
|---|---|---|---|---|
| Flexibility | Strong Support | Limited Support | Strong Support | The atomicity of the elements that make the approach provides flexibility to add and manipulate these elements. The same feature is also generally provided by Web standards. On the other hand, System Metadata tends to be rigidly defined as an exact image of the data model. |
| Decision Taking | Strong Support | Limited Support | No Support | With business rules and forecasting methods integrated in the proposed approach unlike the other ones discussed, decision taking is much more widely supported. |
| Workflow Management | Strong Support | Strong Support | No Support | Based on the presence of data model and system descriptions, Workflow can be managed in the proposed approach and in system metadata, based on triggers watching any events occurring on the related structures. The case is different in Web Metadata. |
| Didactic Properties | Strong Support | No Support | Limited Support | Unlike the rest of the approaches in the way it was clearly presented, the proposed approach has a very imperative instructional property |
| Data Synchronization | Strong Support | Limited Support | No Support | By defining the elements and taxonomies of the systems, the approach reveals to be very beneficial in the synchronization of data between dissimilar systems, outgoing the other ones. |
| Query Builder | Strong Support | Strong Support | No Support | For future query construction engines design or preparation, the proposed approach reveals itself to be even more supportive than regular system metadata approaches, especially with the presence of predefined business areas prepared. The Web Standards are a bit far from providing this feature.. |
| Search Facilities | Strong Support | Limited Support | Strong Support | Search functions are the main purpose behind the creation of Web Metadata, the proposed approach uses derived supporting conceptions to provide search utilities. System metadata however do not address this issue. |

| | | | | |
|---|---|---|---|---|
| Code Generation | **Limited Support** | No Support | No Support | The proposed approach, unlike existing ones, tends to open the door to a probable code generating engine. The approach can be further enhanced to fully support such an important feature. |
| Integration Capabilities | **Strong Support** | Limited Support | No Support | System metadata elements being available, the approach reveals to be very beneficial in the integration of data into available systems, whether initial load or any needed data or information. |
| Change Management | **Strong Support** | Limited Support | Limited Support | The atomicity of the elements, as well as the integration of rules, make the approach very suitable for change management. This feature is generally supported but in a limited way by the rest of the approaches. |
| Report Interface | **Strong Support** | Limited Support | No Support | For future report designing engines design or preparation, the proposed approach reveals itself to be even more supportive than regular system metadata approaches, especially with the presence of predefined business areas prepared. The Web Standards are a bit far from providing this feature. |
| Interoperability | **Strong Support** | Limited Support | Strong Support | The proposed approach, similarly to Web metadata standards, provides a way for information to communicate their existence and their nature to other electronic applications or tools and to permit migration of information between applications or systems. This aspect is not developed in system metadata. |
| Information Communication | **Strong Support** | Limited Support | Limited Support | As an example, documentation capabilities have been conceived in the approach. They tend to encourage all kinds of information communication between company employees for example. |
| Adaptability with Grid Requirements | **Strong Support** | Limited Support | Limited Support | As previously discussed, the approach proposed is convenient to the requirements of Grid Metadata. This is not noticed in the rest of the mostly known approaches and standards. |

| | | | | |
|---|---|---|---|---|
| General Web Requirements | **Strong Support** | No Support | Strong Support | While system metadata are far away from satisfying the general needs of Web metadata, the approach proposed has in common the elements of the Web metadata standards; together they tend to equally provide all facilities required by Web Metadata. |
| DB& Applications Requirements | **Strong Support** | Limited Support | No Support | While Web metadata is far away from satisfying the general needs of systems and database applications metadata, the approach proposed has in common the elements of the system metadata standards; together they tend to equally provide all facilities expected by DB and applications metadata.. |

*Table 7: Comparison of available metadata tools with the proposed Metadata Solution*

## 7.5 Conclusion:

Just like it was clearly stated in the problem definition, the main purpose of the study is to provide an approach that would be suitable for the maximum number of requirements, and that would provide the largest number of facilities and applications. This can be deduced by the above previous section's table that illustrates some of the features and attributes of the proposed information model. Unfortunately, with the lack of the engine relying on this approach, it was not possible to provide accurate measurable quantified results information.

In conclusion, obviously, due to the atomicity and extendibility of elements, and depending on the requirements of the business, the tools used for building the metadata, the model may be enhanced to accommodate even more specific requirements and needs.

# CHAPTER V

# GENERAL CONCLUSION

# 1. STUDY OVERVIEW:

If the word chaos had to be described, one definition of chaos could well be a system without metadata, without a well designed complete and fully comprehensive metadata model. If one has reached to read, conclude, even believe in the previous sentence by the end of this study, and therefore decided to take up the proposed metadata model against a sea of irrelevance, then without any doubt, this thesis has met its objective.

Throughout this study, the detailed features and aspects of what can be an optimal advantageous approach in the grouping, design and modeling of all this metadata have been introduced and gone into; an approach that would provide the most accurate beneficial information to be used at the same time by applications developers and end-users, by web librarians and surfers, all taking into consideration previous standards and conclusions so far developed, whether in the system metadata or the approaches described...

# 2. FUTURE RESEARCH:

Far from denying it, the repository will not prove its efficiency and valuable contributions without a matching engine considered based on its proposed algorithm.

Therefore, schema-based access to standard infrastructure such as parsers, transformation engines, engines that would feed and extract rules and metadata, and even detect some inaccuracies and inconsistencies of metadata, designed in an optimal way, with the most favorable algorithm, would surely be the most challenging, rewarding and continuously promising extension of this study.

# REFERENCES

[1] A Perkins, Business Rules = Metadata http://cdsl.computer.org.dl/proceedings/tools/2000/0774/00/07740285. (2000)
[2] Business Rules Group, Business Rules Manifesto (The Principles of Rule Independence), Business Rules Group. (2003)
[3] DCMI Usage Board, Overview of Documentation for DCMI Metadata Terms, http://dublincore.org/usage/documents/overview. (2003)
[4] DCMI Usage Board, DCMI Metadata Terms, http://dublincore.org/documents/dcmi-terms. (2005)
[5] D-Lib Magazine, Metadata Principles and Practicalities, http://www.dlib.org/dlib/april02/weibel/04weibel.html. (2002)
[6] F. Manola, E. Miller, RDF Primer W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/. (2004)
[7] G. Demarest, Zen and the Art of Information http://www.oracle.com/technology/oramag/oracle/05-mar/o25info.html Oracle Magazine, (March/April 2005)
[8] G. McGovern, Quality metadata makes for successful web content, http://www.gerrymcgovern.com/nt/2004/nt_2004_06_21_metadata.htm. (2004)
[9] HPWREN, High Performance Wireless Research and Education Network, http://hpwren.ucsd.edu/news/011109.html. (2001)
[10]J. F. Koopmann, The Power of Meta-Data http://www.dbazine.com/ofinterest/oi-articles/koopmann6. (2005)
[11]J. Gabriel, Be sure to manage your metadata. http://www.looselycoupled.com/opinion/2005/gabri-metadata-gov0218.html. (2005)
[12] J. Gabriel, Metadata Evolution Management in Your SOA: Managing the life cycle of your metadata. http://webservices.sys-con.com/read/47670.htm. (2005)
[13] J. Greenberg, A. Crystal, Iterative Design of Metadata Creation Tools for Resource Authors, University of North Carolina at Chapel Hill, USA http :// www.siderean.com/dc2003/202_Paper82-color-NEW.pdf. (2003)
[14]L. Guerts, J. Hunter, The dynamic generation of intelligent multimedia presentations through semantic inferencing. http://archive.dstc.edu.au/maenad/ecdl2002/ecdl2002.html. (2002).
[15]Library of Congress, METS (Metadata Encoding and Transmission Standard). http://www.loc.gov/standards/mets/. (2003)
[16]M. Gunderloy, J. Jorden, Mastering SQL Server 2000, Sybex. (2000)
[17]P. M.D.Gray, L. Kerschberg, P. J.H.King, A. Poulovassilis, The Functional Approach to Data Management, Modeling, Analyzing and Integrating Homogeneous Data. Springer-Verlay Berlin Heidenberg. (2004)
[18]R. S. Seiner, TDAN Curt Hall - Problems with Conventional Metadata. The Data Administration Newsletter. http://www.tdan.com/i008ht02.htm. (2003)
[19]RoMEO, Project RoMEO (Rights MEtadata for Open archiving). http://www.lboro.ac.uk/departments/ls/disresearch/romeo. (2003).

[20] S. Mittra, Database Performance Tuning and Optimization, Springer. (2003)
[21] S. Patil, E. Deelman, G. Singh, M. Atkinson, A. Chervenak, N. Chue Hong, Carl Kesselman1,L. Pearlman1, Mei-Hui Su, Grid-Based Metadata Services. http://csdl.computer.org/dl/proceedings/ssdbm/2004/2146/00/21460393.pdf
[22] Stanford Encyclopedia of Philosophy, Albert Einstein - Philosophy of Science http://plato.stanford.edu/entries/einstein-philscience/#3. (2004)
[23] T. Chan, M. Sharples, G. Vavoula, P. Lonsdale, Educational Metadata for Mobile Learning http://csdl.computer.org/comp/proceedings/wmte/2004/1989/00/19890197.pdf. (2003)
[24] T. Reamy. To Metadata or Not To Metadata http://www.econtentmag.com/Articles/ArticleReader.aspx?ArticleID=7118&CategoryID=21, EContent Magazine - October 2004 Issue, (Posted Oct 15, 2004)
[25] W3C, RDF Vocabulary Description Language. http://www.w3.org/TR/rdf- schema/. (2003)
[26] W3C, Web Services Description Language (WSDL). Version 1.2, http://www.w3.org/TR/wsdl12. (2003)
[27] Wipro Technologies, Customized Metadata Solution for a Data Warehouse - A Success Story http://www.dmreview.com/registration/index.php?refer=%2Fportals%2Fwhitepaper%2Ecfm%3FtopicId%3D230011%26WhitePaperID%3D10573&CFID=7730208&CFTOKEN=43742914. (2003)

# APPENDIX

## ***Business Rules Manifesto (The Principles of Rule Independence)***



Rules are a first-class citizen of the requirements world.

Rules are essential for, and a discrete part of, business models, system models, and implementation models.

Rules are not process and not procedure. They should not be embedded in either of these.

Rules build on facts, and facts build on concepts as expressed by terms.

Terms express business concepts; facts make assertions about these concepts; rules constrain and support these facts.

Rules must be explicit. No rule is ever assumed about any concept or fact unless a rule has been specified explicitly.

Rules are basic to what the business knows about itself - that is, to basic business knowledge. Rules need to be nurtured, protected, and managed.

Rules are about business practice and guidance; therefore, rules are motivated by business goals and objectives and are shaped by various influences.

Rules should be expressed declaratively for the business audience, in natural-language sentences. If something cannot be expressed, then it is not a rule.

Rules are best implemented declaratively. Rules are based on truth values. How a rule's truth value is determined or maintained is hidden from users.

Rules are explicit constraints on behavior and/or provide support to behavior.

| |
|---|
| Rules generally apply *across* processes and procedures. There should be one cohesive body of rules, which should be enforced consistently across different areas of business activity. |
| The relationship between events and rules is generally many-to-many. |
| A rule statement is distinct from the enforcement level defined for it. These are separate concerns. |
| Rules should be defined independently of responsibility for the *who*, *where*, *when*, or *how* of enforcement. |
| Rules often require special or selective handling of detected violations. Such rule violation activity is activity like any other activity. |
| Rules define the boundary between acceptable and unacceptable business activity. |
| To ensure maximum consistency and reusability, the handling of unacceptable business activity should be separable from the handling of acceptable business activity. |
| Exceptions to rules are expressed by other rules. |
| Rules always cost the business something. This cost must be balanced against business risks. |
| Rules should arise from knowledgeable business people. |
| Business people should have tools available to help them develop and manage rules. |
| In the long run, rules are more important to the business than hardware/software platforms. |
| Rules should be managed in such a way that they can be readily redeployed to new hardware/software platforms. |
| A business rule system is never really finished because it is intentionally built for continuous change. |
| Rules, and the ability to change them effectively, are key to improving business adaptability. |

*Table 8 Business Rules Manifesto*

***End of the Business Rules Manifesto document [2].***