Tool Support and Data Management for Business Analytics

Mana Azarm

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the M. Sc. degree in Systems Science

University of Ottawa

Ottawa, Ontario, Canada

May 2011

© Mana Azarm, Ottawa, Canada, 2011

Abstract

The data delivery architectures in most enterprises are complex and under documented. Conceptual business models and business analytics applications are created to provide a simplified, and easy to navigate view of enterprise data for analysts. But the construction of such interfaces is tedious, manually intensive to build, requiring specialized technical expertise, and it is especially difficult to map exactly where data came from in the organization.

In this paper we investigate how two aspects (lineage and requests for data i.e. semantics and new reports) can be addressed by tying metadata documentation to a systematic data delivery architecture in order to support business analytics applications. We propose a tool framework that includes a metadata repository for each step in the data delivery architecture, a web based interface to access and manage that repository and mapping tools that capture data lineage to support step by step automation of data delivery.

Acknowledgements

To begin, I wish to thank my supervisor Dr. Liam Peyton for all his time and effort to guide me through this work. I owe him everything I know from a year and a half under his supervision. Without his thorough and well scheduled supervision, I would have never been able to accomplish my Master's degree. He not only taught me about modern science and technology but also taught me how to be a professional.

I thank the Ottawa Hospital for giving me the opportunity to observe their procedures and problems. I want to thank the data stewards at the hospital who trusted me and put confidence on my work.

I also want to thank Fatemeh Nargesian for the time she spent on transferring her experience and knowledge to me. Every meeting with her taught me a great deal and gave me new ideas.

I extend thanks to my family, who supported me my whole life, and never missed a chance to help me towards success.

Finally, I would like to thank my boyfriend Jean-Philippe Daigle, for his help reviewing this thesis, and his love and support throughout my studies.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
BI	Business Intelligence
BPM	Business Process Management
CDL	Conceptual Definition Language
DBA	Data Base Administrator
DW	Data Warehouse
FM	Framework Manager
KPI	Key Performance Indicators
MDL	Mapping Definition Language
MRSA	Methicillin-resistant Staphylococcus Aureus
OLAP	On-Line Analytical Process
SDL	Store Definition Language
UI	User Interface
XML	eXtensible Mark-up Language

Chapter 1. Introduction

1.1. Problem Statement

Nowadays, there are various high tech companies who offer powerful reporting and performance measurement solutions. Although the generation of pleasant graphical reports is not a difficult task anymore, documenting the semantics of data and technical expressions within a report is of utmost importance, yet is almost being ignored in those applications. The lineage of the knowledge produced on the report, and an understanding of how the report elements relate to the actual data elements in the original operational data sources is lost.

When a manager looks at a report, they want to know exactly what each element or technical expression on a report means, where the values shown originate from and how often they are getting updated. Only then will the data in a report make sense. Also, managers or different organizational actors who might not necessarily have the technical knowledge of software development need to be empowered to be able to generate their own customized reports, slightly different than those provided by professional developers.

Business analysts, in particular, need to understand precisely the data lineage of the information displayed on reports in order to ensure accuracy and relevance. An analyst should understand precisely what each number on a report means. This includes understanding where the number came from in terms of the calculations performed on data items as data is moved and transformed across the complex data architecture in today's enterprises, from operational data sources to data warehouses to specialized business analytics applications from which reports are generated.

The reports are the results of a multilayer data architecture which we refer to as the enterprise data architecture. However, the data delivery architecture in most enterprises is complex and under documented. The corporate data warehouse, as one of the layers of the enterprise data architecture, provides a comprehensive space for integrated data throughout the whole organization. In fact, a data warehouse combines data coming from various operational data sources, which are the data entry points of the different departments or branches of an organization.

While the operational data sources operate every day, they inject more data to the system each day. As the volume of data in the system and data warehouse increases, the architecture gets more complicated and therefore the bonds of data among different layers of the enterprise data architecture become looser and vaguer, meaning that we are not able to follow the flow of the same piece of data through different layers of the architecture. Typically, conceptual business models (e.g. data marts) and business analytics applications (e.g. dashboards) are created to provide a simplified, and easy to navigate view of enterprise data for analysts.

In order, to make sense of this, metadata is needed to understand the data in an enterprise. Metadata is descriptive data about the data stored in a data warehouse (Inmmon 2005). Metadata documentation includes the availability of the data described by metadata, how these data can be accessed, how data sources are organized and maintained, as well as the documentation of the columns and tables in the data warehouse from which report elements originate. Metadata documentation also provides for indexing the metadata collections for rapid querying and generation of statistical values (Batcheller 2008).

However, the construction of such interfaces is tedious, labour-intensive to build, requires specialized technical expertise, and it is especially difficult to map exactly where data came from

in the organization. In this paper we investigate how both aspects (lineage and meaning of report elements) can be addressed by tying metadata documentation to data delivery architecture in a systematic manner in order to provide managers and other end-users with a better understanding of the data they are interacting with in business analytics applications.

In particular, we have focused on the particular needs of hospitals and other healthcare organizations in our case study work. Healthcare is especially relevant as an application domain for our work because of the critical nature of the data shown in reports. For example, when a healthcare provider looks at reports to manage quality of care, they want to understand what each number means and where it came from. For instance, when viewing the infection control indicator shown on a report for January 2009, how is total patient population calculated? Does it include outpatient treatments? Does total MRSA include percentage of patients who were transferred to select nursing units and had an MRSA screening test ordered within 24 hours with a positive result?

Healthcare organizations are also representative of large enterprises and their complex data delivery architectures. The hospital in our case study has over 10,000 employees. The data related to a typical large healthcare organization are shared among diverse organizations [29] and health-related data arrives from various related organizations and their operational data sources such as clinics, laboratories, physician's offices, emergency rooms, intensive care, operating rooms, pharmacy etc.

1.2. Thesis Motivation

In recent decades, business analytics, supported by an enterprise data architecture is critical to any organization that wants to ensure that they embrace a culture of continuous

improvement (Williams and Williams 2004). Business analytics takes collected data from sources throughout an organization as well as external sources, and presents it in views that create knowledge for decision making (Lonnqvist and Pirttimaki 2006). Reports are generated from these business analytics applications.

However, there are significant challenges that are faced today in supporting such business analytics applications. A literature review and our experiences with our healthcare case study show that there is a large cognitive gap between the sophisticated data architecture that a handful of technical experts can provide an enterprise, and the ability of thousands of end users to understand the data in the reports delivered by that architecture or to even understand what reports they could be requesting.

The motivation for this thesis is to develop an approach along with a toolset to bridge that gap. These gaps fall between different layers of the data architecture which is composed of operational data sources, ETL layer, data warehouse, conceptual business model, and business analytics applications.

A tool supported meta-data repository in the context of an enterprise data architecture is needed to automate navigating lineage from the reports back to models and to the data warehouse by finding and documenting the forgotten links between different layers of the architecture. Moreover, we try to provide knowledge to report authors and design web tools for them to access and edit metadata and link it to the ability to provide dynamic documentation of reporting elements that end users interact with.

1.3. Contributions

Metadata documentation is critical in order to enable end-users to understand the meaning of the numbers they see in reports, and to be able to communicate effectively the reports they want to see. The most natural place to organize and collect metadata documentation is to create and support a metadata repository alongside a central data warehouse where data is collected and consolidated across an organization.

To resolve the issues related to enterprise data architecture and the long distance from numbers on a report to data elements within a data warehouse, we take existing conceptual business models that support business analytics applications based on star schema (Inmon, 1996), multi-dimensional cubes (Cognos, 2008), or more recent research initiatives (Nargesian, et al. 2010) and extend them with a mapping infrastructure.

The mapping infrastructure is used to document the data lineage from report elements through the entire data delivery architecture back to the operational sources from which the data originated. We illustrate and evaluate our approach with a prototype implementation and a case study using a health care analytics dashboard for managing hospital-acquired infections.

The thesis contributions are:

1. A gap analysis of industrial practice and academic research in enterprise data architecture to analyze how well current approaches provide systematic documentation of the data presented in reports so that end users can understand reports and request new ones. The gaps are articulated in terms of evaluation criteria that proposed approaches should meet.
2. A framework to provide and link systematic user-managed documentation for data management across the entire data architecture for business analytics based on

a) a metadata repository that documents the data centrally stored in a data warehouse (including documentation of sources for the warehouse)
b) A systematic set of structured mappings across the layers of an enterprise data architecture that link from report elements to conceptual business models to the central data warehouse from which the data for the report element originated.

3. Tool Support for the framework that enables users to manage and use up to date documentation to understand data displayed in reports including
   a) A web-based interface to access and manage the metadata repository and view the documentation there including the sharing of comments between users.
   b) Dynamic synchronization between metadata in the repository and data in the data warehouse including range and usage statistics.
   c) User interface support for clicking from report elements directly to metadata documentation web application that leverages the structured mappings in 2b).

Some of our results are published in the following paper:

M. Azarm, L. Peyton, F. Nargesian, "Managing and Mapping Data Lineage for Business Intelligence and Analytics Applications in Health Care", IEEE Int'l Conference on Information Society, London, UK, June 2011.

1.4. Thesis Methodology and Organization

The methodology to develop this thesis is based on design-oriented research (Hevner, et al. 2004) category. Design-oriented research seeks to solve or improve a problem in current practice (Fallman 2003). It includes five stages of finding and defining a problem, framework

design, framework evaluation, revaluation and improvement of framework, and communication and discussion of research (Bell , et al. 2007).

Hence, we have employed the design-oriented research guidelines to discover an effective solution to the aforementioned problems. Following the five stages of a design-oriented research, we start with problem definition and then gap analysis. Then we propose our framework and toolset and implement it in an iterative cycle. The evaluation of the proposed approach is carried through a case study as well as a set of proposed evaluation criteria. Then, re-evaluation and iterations of the proposed approach, and discussions on future work follow.

Overall, we have followed the steps below in our thesis:

1. Problem identification and problem definition
2. Establish criteria for evaluation of the proposed approach
3. Selected a representative case study to instantiate our solution
4. Literature review and background research in the related areas
5. Gap analysis based on the literature review and the selected case
6. Development of the solution and proposed approach
7. Implementation of the proposed method on the case study
8. Evaluation of the results of the proposed approach on the case study
9. Iteration of the steps 6-8 to reach the desired results based on our proposed evaluation criteria
10. Discussions about the results of the thesis effort and outlining the future work

The thesis is organized as follows: In Chapter 2, we go over the literature and previous works in related fields. Definitions and explanations of critical concepts and expressions in the

field are also provided in this chapter. We also describe some best practice applications and frameworks that are currently in use.

Chapter 3 describes our proposed framework and associated tool support, but first begins with a clear analysis of the problem and summarizes our gap analysis in terms of the evaluation criteria we are trying to address.

In chapter 4, we describe our case study which is drawn from the healthcare sector. The proposed framework is implemented and applied to the case study in healthcare industry.

Chapter 5 presents an evaluation and assessment of our proposed framework. A comparison between our framework and current practices is presented.

Finally, Chapter 6 summarizes the results and findings of this research thesis and introduces possible areas for further development and research.

Chapter 2. Background

2.1. Knowledge Management

Knowledge management is known to be the presentation, exchange and transfer of knowledge required by an organization to succeed. Data as the foundation of knowledge needs to go through the hierarchical transformation depicted in Figure **2-1**. The difference between knowledge and information are the attitudes, systems and skills to retrieve information and share it in a new context (Dubois and Wilkerson 2008).

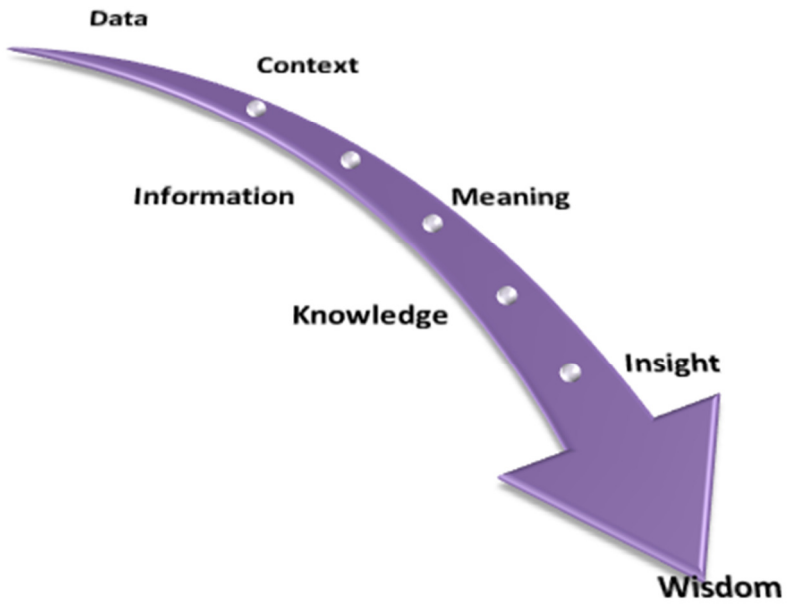

Figure 2-1: Hierarchy of Knowledge

Good knowledge management will require support from technology. Technology should be focused on the enabler tools search and retrieve knowledge efficiently (Scammell 2001). Knowledge management portals such as business intelligence tools support data warehouses, routine data analysis, report writing, analytical processing, and data mining (Rao 2005).

2.2. Enterprise data architecture for Business Analytics

Business analytics in the context of an enterprise data architecture spans from operational data sources to data warehouses to conceptual business models (star schema, multi-dimensional cubes etc.) to business analytics applications (including ad-hoc queries and reports). Analytics applications in the past have been mostly architected with technologies accessing data warehouses directly, but nowadays they have evolved towards distributed multi-tier analytic applications. (Liya, Barash, & Bartolini, 2007). A typical architecture for business analytics includes the following layers (ascending):

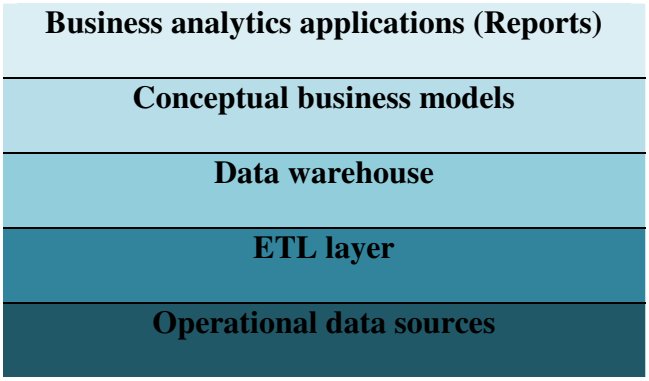

Figure 2-2: Architecture for Business Analytics

Operational data sources are where data from transactional systems are stored. These are the entry points for data into the enterprise data architecture. Organizations have multiple source systems which might be very different technically and therefore store different data types and formats (Han & Kamber, 2006).

The ETL layer involves the extraction of data from operational data sources and transforming the data into a standard compatible format that is used to load the data into a central data warehouse where a complete integrated data view of the organization is stored. Typically, a data warehouse is the centralized repository of data within an organization and provides the basis

for most business analytics applications (Kimball, Ross, Thornwaite, & Mundy, 2008). However, there can be more than one data warehouse in an organization and one can build business analytics applications directly off of operational data sources and external data as well.

A conceptual business model provides a meaningful model that understands the relationships between data entities stored in a data warehouse. This layer can also be called the business logic model, as it presents a logical view of the entities independent of the implementation platforms or technologies (Liya, Barash, & Bartolini, 2007). It includes multidimensional data marts that can be generated to address focused analytics requirements (Leitheiser, 2001).

In conceptual business models we often encounter data marts. A data mart is a business analytics application that bundles reports and analysis around a focused multi-dimensional view of a subset of the data from the data warehouse. Each data mart is represented by a dimensional model and this model is built from a conformed fact table and conformed dimensions (Dayal, et al. 2009). Aggregates are often pre-computed hierarchically along each of the dimensions in order to improve performance for reports and enable flexible "drill up" and "drill down" navigation (e.g. to drill down and see totals by country, then by province, and finally by city).

On the other hand, OLAP tools handle multidimensional data marts for analytic purposes. OLAP tools are specifically applicable to processing large amounts of data to come up with the analytic analysis (Dong, et al. 2004). And finally, business analytics and visualization functions and applications utilize OLAP cubes in the processing and formatting of data for analysis and display (Poole 2001).

Business analytics applications include any software application that uses data in the data warehouse to produce meaningful reports or to measure performance indicators through ad-hoc querying, on-line reporting or on-line analytical processing. These applications publish reports that analyze operational data according to user specific requirements (Cognos 2008).

Figure **2-3** shows a complete flow of the data through the layers of the enterprise data architecture. Real world events are injected into the system through the operational data sources which translate them into the predefined schemas in order to fit them in the relational structures and tables. Then, the operational data from different sources are manipulated by the ETL system to be stored in the enterprise wide data warehouse which accumulates and integrates corporate data (Inmmon 2005).

In order to fulfill reporting needs, individual conceptual business models are designed to accommodate and restructure relevant data to a report into dimensional schemas. Then the conceptual business models are employed by a business analytics application to produce desired reports.

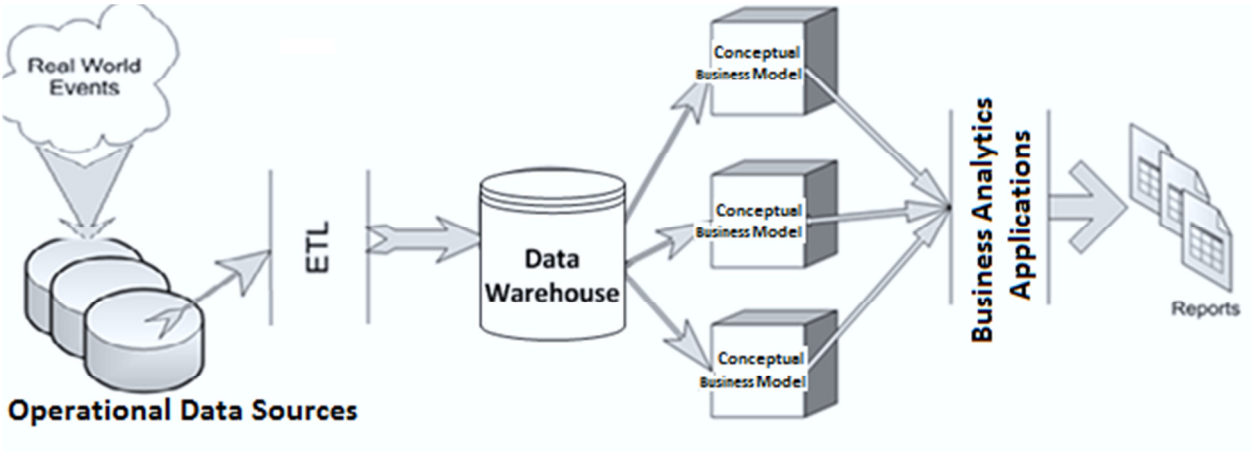

Figure 2-3: Enterprise data architecture

2.3. Business Analytics

Business analytics involves the collection, analysis and reporting of data in order to monitor business processes and evaluate key performance indicators and indexes in order to provide insights into the business performance of an organization. Various business analytics applications have been developed to aid stakeholders observe their processes in a detailed manner, know the related data items, generate periodic reports about metrics and indicators, and analyze data values (Azvine, Nauck and Ho 2003).

A managerial decision requires relevant information and operational data from operational data sources to be collected, integrated and analyzed. On-line analytical Processing (OLAP) and data mining techniques are usually employed to perform business analytics. Data mining has been widely used to carry planning, forecasting, monitoring and controlling tasks and procedures (Berendt, Preibusch and Teltzrow 2008).

On the other hand, the way data is stored in tables in traditional relational data bases does not help business analyzers in organizations. The way they think is far away from relational tables. They need a multi-dimensional flexible view at a conceptual level in terms of business concepts so that they can apply them in multiple reporting requirements and scenarios. OLAP tools are based on multidimensional concepts. There are some rules governing OLAP applications (Meier, Sinzig and Mertens 2005):

a) Multidimensional conceptual views: as an analyst pictures the data, they can see both facts and dimensions representing each aspect of analysis. Facts are quantitative measures like price, cost, weight, duration etc. Examples of

dimensions are time (Year, month day), location (country, province, city), products (sporting goods, golf, golf club), etc.

b) Spontaneous data processing: data presented in charts produced from a dimensional application should have the capability of drill through. It means that we can go deeper into more details e.g. time dimension can be drilled though year, quarter, month and day to demonstrate more specific details.

c) Access to low level operational data: entails the management of data coming from multiple source systems and low-level transactional applications (Vercellis 2009).

d) Separation of OLAP data and operational source data: the dimension and fact related info in dimensional modeling should be generated from the operational source data and be kept in separate repositories (Chaudhuri and Dayal 1997).

e) Differentiation between missing values and zeros. This feature is especially important in statistical analysis. For example, if we are counting the number of patients that have a Community Acquired Methicillin Resistance in a column that could take a zero or one Boolean value, we cannot consider a field that hasn't been filled up as zero because zero means that person was not Community Acquired Methicillin Resistant while a null value might indicate that the person has not been tested for Methicillin Resistance index and therefore we don't know if they are resistant or not. Hence, it's important that a business analytics application differentiates the null and zero values.

Traditionally, business analytics applications support decision makers with applications that are periodically fed from operational data sources i.e. they are not responding to changes instantly. More recently, there have been enhanced business analytics applications which are

continuously being fed from operational data sources so that they can reflect the latest changes and generate the most up to date values and analysis (Seufert and Schiefer 2005).

2.3.1 Enterprise Reporting

Enterprise reporting systems organize and present data in various formats in order to accommodate the various requirements and needs of users and managers (Han & Kamber, 2006). On the other hand, enterprise reporting goes well beyond simple queries and reporting documents. It in fact includes tools for visualizing data. Enterprise reporting is provided by tools as simple as ad-hoc SQL query tools or as advanced as industry tools such as the IBM Cognos product line (Gibson, Arnott, & Jagielska, 2004).

The Executive Dashboard is a special type of enterprise report that is typically envisioned as the senior executive's window into the business. It's a collection of key measures (KPIs, BPMs, and so on) that are meant to tell the executive the current state of the business and its general direction (Few 2006).

On the other hand, dashboards can bring great values to an enterprise. However, providing data for dashboards is not an easy task. To provide for a single data item on a dashboard we might have to look at and integrate various and often inconsistent data repositories or data sources.

2.4. Data Warehouse

A data warehouse is used to collect and store a complete, consistent and integrated view of data from all areas of the organization. A data warehouse should (Inmmon 2005)

- make an organization's information accessible
- present the information consistently

- be adaptive and resilient to change
- be a secure bastion that protects the information
- serve as the foundation for improved decision making
- be accepted by the business community

Data warehouses supply information from the transactional systems to reporting and knowledge discovery systems (Seufert and Schiefer 2005). The construction of data warehouses involves data cleaning, data integration and data transformation and can be viewed as an important pre-processing step for data mining. Data warehouses also provide the foundation from which on-line analytical processing (OLAP) tools support interactive analysis of multidimensional data of varied granularities.

A data warehouse system usually requires the following components (Kimball & Ross, The Data Warehouse Toolkit, 2002):

- Operational data sources and ETL: As mentioned earlier in the data architecture section, operational data sources enter the corporate data into the data architecture. Data warehousing systems as a sub division of the enterprise data architecture include the operational data sources.
- Data presentation: is where the data is stored and made available for querying. In fact it's the integrated and comprehensive data repository of a warehousing system.
- Data access tools: Can be as simple as an ad hoc query tool or as complicated as a sophisticated data mining or modeling application. It's the data access tools that can provide answers to measurement indices and performance indicators. The

queries are written in this environment and can even supply the metadata info. One example could be SQL Server query designer.

- Metadata: a detailed description of metadata follows.

2.4.1 Metadata

As mentioned before, metadata can be considered as a part of a data warehousing system. Because of the importance of this concept to our thesis attempt, we decided to dedicate a separate section to it.

Metadata describes the context by explaining the meaning to data warehouse contents (Inmon 1996). "Metadata has been identified as a key success factor in data warehouse projects" (Vetterli, Vaduva and Staudt 2000). Metadata maximizes the exploitation of the data warehouse by providing the meaning to the data warehouse component.

Metadata can be classified into two categories: business metadata and technical metadata. Business metadata is appealing to end users and consists of end-user-specific documentation, domain-specific ontological knowledge, specialized expressions and terminology, and dictionaries. On the other hand, technical metadata includes schema description, configurations, physical storage info, runtime information and log files, data execution specifications, and security issues (Vetterli, Vaduva and Staudt 2000).

2.5. Conceptual Business Models and data marts

There are a variety of technologies and concepts whose names are used interchangeably when we talk about conceptual business models. In this section, we discuss all of these, but will refer to them generically throughout the rest of the thesis as "conceptual business models".

A data mart allows data to be modeled and viewed in terms of multiple dimensions and facts. A multidimensional data model is typically organized around a central theme which is represented by a fact table. We consider data marts as equivalent to multidimensional OLAP (MOLAP) cubes (Chaudhuri and Dayal 1997).

In the structure of data marts we can find fact tables and measure tables. The fact tables are created in response to a business measure (performance measures or indicators) (Malinowski and Zimanyi 2008) such as amount of sales for a car manufacturer which indicates if the company has reached their sales target. Since fact tables represent a performance measure, a row in a fact table corresponds to a measurement e.g. daily sales fact table. Fact tables are deep in the number of rows but narrow in the number of columns.

Dimension tables provide business context to the facts. They have many columns or attributes but are shallow in the number of rows. There are four steps for dimensional design (Busborg, Tryfona and Christiansen 1999):

1) Selection of the business process to model
2) Declaration of the grain of the business process: it means specifying exactly what an individual fact table row represents.
3) Choosing the dimensions that apply to each fact table row
4) Identification of the numeric facts that will populate each fact table row.

2.5.1 Conceptual Definition Language

A recent approach to conceptual business model is conceptual definition language (CDL) (Nargesian 2010). A correspondent CDL is defined in which the dimensional reporting elements are represented in an XML format as a conceptual model (Conceptual Definition Language) on top of the underlying data warehouse (modeled as Store Definition Language).

The relation of the conceptual model and data warehouse is provided by a set of mapping correspondences. Looking at a report as a cube with measures and hierarchies of descriptive data, each report can be documented by considering the portion of CDL which models the particular measures and hierarchies. This way, the sources of data in the report can be traced by the mappings in Mapping Definition Language. Therefore, our metadata repository and its correspondent conceptual model provide the data lineage between reports and data warehouse (Rizzolo, et al. 2010).

Hence, in this approach any information on a report has a corresponding semantic construct in CDL that can be linked to the data warehouse. Moreover, it is mapped to a cell in the dimensional data mart and then to one or more columns, procedures or fields in the data warehouse (Duan, Chen and Li 2010). CDL describes the dimensional data elements i.e. fact table, dimensions and their attribute in mark-up language in our proposed framework.

2.6. Data Lineage and Report Documentation

There has been ongoing work in progress to address the problems concerning the lineage of reports and analytical views to the source data which is mostly causing incomprehensible reports in which the elements have lost their roots back to the data warehouse and operational data sources.

Most of the previous studies in this regard consider finding the correspondent algorithms and their automatic generation for all the entities in a warehouse. However there have been a few studies which involved the development of practical and user friendly software applications that employ the algorithms towards demonstrating lineage to stakeholders (Duan, Chen and Li 2010).

On the other hand, in other work, lineage involves tracing the warehouse data items back to the operational source systems (Cui and Widom 2003). However, the lineage from the data warehouse to the data marts and conceptual business model and then to reports is the part that has been paid the least attention. These documentations provide meaning and links to the building blocks of a report (Rizzolo, et al. 2010).

As mentioned earlier, the documentations help data stewards and corporate executives to fully understand the reports about business measures. Again, there hasn't been much work on documentation of the lineage considering report elements. What we can mostly find is the lineage between the data warehouse and operational data sources (CUI and WIDOM 2000).

2.6.1 Lineage in a relational world

There have been studies towards formulating the lineage problems by giving declarative definitions of the lineage for complex relational views, developing the tracing algorithms and showing how to perform lineage tracing in an efficient manner (Pin and Chen 1976). In the spectrum of this thesis, means to extract relational lineage can be applied to the lineage between the source system data and data warehouse entities.

There are multiple steps to developing tracing algorithms in a relational perspective. First, a set of base relation tuples in a relational data warehouse that produce a given view is identified. This is called tuple derivation. Then tuple derivations for operators and views are defined. These studies define classes of views over base relations using the relational algebra operators like join, set, union, and etc.

Considering a view as an operator tree, tuple derivations for views are defined by defining a set of all the base tuples that contribute to form a view. Then derivation tracing queries are defined using relational queries over the base data. Subsequently, lineage tracing algorithms are developed for relational views with aggregation. These algorithms identify the exact set of base data that produced a view data element (CUI and WIDOM 2000).

2.6.2 Lineage in a Dimensional World

In the dimensional world, the lineage includes dimensional entities and the lineage between reporting elements generated by business analytics applications and conceptual business models as well as the lineage between the conceptual business models and data warehouse. In this sense, the correspondent maps should describe the lineage of the fact table to the elements on a report and also to the entities inside the data warehouse (Eder and Koncilia 2001).

When a report is linked to the conceptual dimensional elements, correspondents between the dimensional elements and data warehouse entities are well defined, and data warehouse entities are well linked to their source systems, we can claim that we have a comprehensive documentation which covers the whole spectrum of the enterprise data architecture from the very bottom source applications to the very top level reports (Rizzolo, et al. 2010).

2.7. Metadata documentation web application tools

To have a thorough grasp of data in a warehouse which combines different sorts of data from different and various source systems, we need to document it. Metadata is known as “all the information that defines and describes the structure, operations, and contents of a data warehouse or business intelligence system” (Kimball, Ross, Thornwaite, & Mundy, 2008).

The metadata increases the speed and possibility of retrieving documents or pieces of data and helps the control and management of data (Holzinger, Kleinberger and Müller 2001). In particular, for our thesis, we focus on descriptive knowledge about data entities, lineage, statistics, quality, access rights etc. as well as semantic understanding, comments, communication and coordination.

Documentation tools document and present the metadata knowledge in a clear, coherent and easy to navigate format. Any documentation effort in this area should identify the table and column names, definitions, and also calculation rules and their attributes (Breault, Goodall and Fos 2002).

Metadata documentation tools often include a metadata repository which stores data warehouse entities, descriptions and information and can be updated in real time in an integrated manner (Han & Kamber, 2006) through the documentation tool. Metadata repository can be a SQL database with a relational schema.

As a matter of fact, a metadata repository reflects real time changes of the data warehouse and the application shows them instantly as well. This architecture enforces a dynamic mechanism for knowledge discovery which is very important to users.

Metadata documentation tools are either generated by a human who is usually a professional metadata creator or generated automatically through machine processing which are often web based. This feature allows users to be able to view the content of the data warehouse regardless of their physical location. Metadata generation tools include intellectual tools, metadata standards, and technical compilations (Greenberg 2003).

There are basically two categories of metadata management tools that can be considered: general-purpose versus model-based tools. The first category is general-purpose metadata

management tools that aim at the enterprise wide data and tries to document the whole system. These tools are meant to document and archive structures, systems and applications (Vaduva & Dittrich, 2007). General purpose metadata tools are not successful on their own. They require to be integrated with some other applications to be able to update data in the repositories and to maintain development purposes.

The second category of data management tools are metadata driven tools. These tools use metadata to achieve specific tasks of which we can point at building the data warehouses. These tools are distributed between the data repository and the software engine which come to participate together at runtime (Vaduva & Dittrich, 2007).

Chapter 3. Tool support and data management for business analytics

3.1. Problem Description

When a Healthcare manager looks at a report, they want to know exactly what each element or technical expression on a report means, where their value is fed from and how often they are getting updated. On the other hand, the manager needs to be aware of all the items that are available in a report to generate new reports by putting them together through a reporting application..

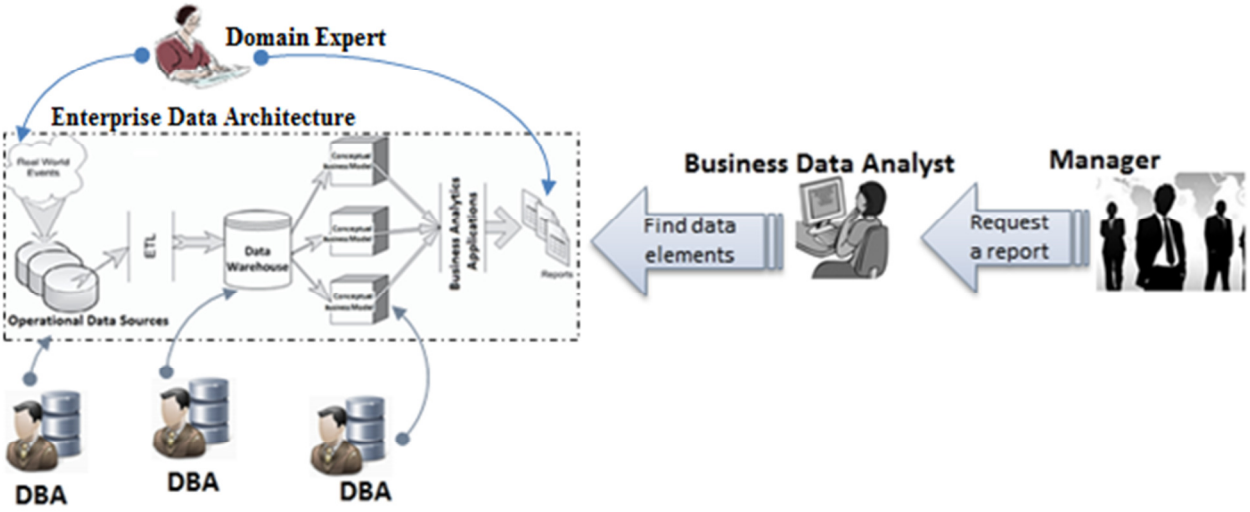

Figure 3-1: Actors involved in the data lineage problem

Figure **3-1** shows a schematic picture of the actors who can be involved in a reporting attempt in the current enterprise data architecture. In current practice, when a manager needs a special report, the data analysts are asked to provide the required documents. Depending on the content and requirements of the report, the analyst or a team of analysts should spend quite a

while to dig through the various layers of data architecture to pull out the required data. Then, they have to apply the right set of functions and formulas to generate the required reports.

The first step to generate the reports is to identify the required data items and how they are contributing to the results. Then, we need to know where to find those specific data items and what data columns from which databases in what system are feeding them and how often they are being updated.

Data analysts have been contributing to the last two layers of the architecture. They usually know about the conceptual business models and they have used those models in business analytics applications. However, this could only happen when they knew the correct data elements for a report and where they originate from.

But, within the current complicated architecture, analysts alone do not have the knowledge or required expertise to extract data from multiple layers of the architecture. They usually call for help from DBAs and domain experts. The DBAs are the experts in databases but there are different DBAs for different layers of the data architecture such as operational data sources, data warehouses, and data marts.

On the other hand, domain experts are the ones who know about the events in the operational data source systems and also the business analytics applications and reports. Therefore, they know about the applications at both ends of the architecture. Yet, they usually don't understand how data is getting transformed through the different layers of the architecture. Also, they are not aware of the ongoing changes of data and its flow in the architecture.

As a matter of fact, in current practice the generation of one specific report may require the union of four or five different domain experts. This union is not cost efficient as the experts

leaving their regular agenda impose an expensive salary cost for multiple meetings as well as the time they spend to provide required results.

In the current practice, generation of a report forms an ad-hoc project which gathers multiple experts. The results of such a project in the best scenario are documented in scattered documents in a file based storage system, static simple HTML pages, dispersed and isolated databases which are accessible only via their own creators.

These documentations form another source of disintegrated knowledge which is very difficult to reach, manage and discuss. Also these islands of documents being disconnected and isolated get out of date very quickly as opposed to the rapid changes of form and content of data in the architecture. As a matter of fact, data stewards were seeking for tools and approaches to help them have a better system of documentation and data tracking along with means of communications to retain and maintain their efforts for future applications.

3.2. Evaluation Criteria

Our thesis is focused on addressing the issues outlined in section 3.1 above. Based on a gap analysis from both our literature review and our healthcare case study, we have derived the following list of evaluation criteria to articulate the features that should be present in order to address those issues. In chapter 5, we will use these criteria to evaluate how well our approach addresses those features in comparison with other approaches. This is preliminary design-oriented research where the intent is to demonstrate the potential of our approach.

The study of current industry practice of enterprise data architecture involved direct interaction with the local hospital data analysts and DBAs. The gaps were recognized by studying the requirements of the professionals and ways to facilitate and ease their daily agenda.

The proposed approach evolved through weekly meetings with the hospital data analysts. Each meeting along with further literature review led to the development of new features and solutions.

3.2.1 Metadata documentation

The first set of criteria seeks to assess the role of metadata for documentation efforts. This means that our documentation revolves around metadata knowledge. Comprehensive metadata documents should provide the right descriptions and explanations (meaning and lineage) along with quantitative analysis and critics about the components (statistics and comments). Below, we can see the four criteria in this regard:

a. Meaning: provide an understanding of a report context and how often it is updated. This criterion seeks to examine how well the data elements of the architecture are described in order to help each of the stakeholders fully understand them.
b. Statistics: The ability to provide live statistical analysis over the values recorded in each column of the data warehouse. Statistics provide a better grasp of the situation. Therefore, we want our documentation to be equipped with relevant statistical indexes and values.
c. Lineage: The availability of a clear path between the report elements and their roots in data warehouse. We want to measure how well this path is recognized towards filling the gaps between different layers of the enterprise data architecture.

d. Comments: to facilitate communications and means to share comments and discussions. As an example, we encourage if the documentation effort provides for sharing comments about each specific data element in the most convenient place.

3.2.2 Availability of documentation

These criteria evaluate the accessibility and ease of reach to the documentation. We need a mechanism to provide easy and quick access to the documentation for all the stakeholders and experts involved regardless of their physical location.

e. Who: the number of people who are able or enabled to extract and make use of the meaning, stats, comments and lineage. Specifically, by this criterion we measure the number and diversity of the people who can access the documentation.
f. Where: The availability of documentation regardless of where the users are located geographically is important in a smooth and pervasive documentation effort. Also all the documentation should be concentrated in a well-structured system.
g. When: Up to date linage and info in synch with the latest changes in DW to report.
h. How: This criterion investigates the methods to access the meaning and lineage documentations e.g. if the mechanism to extract meaning and lineage is automated or manual?

3.2.3 Documentation effort

Time and cost are the two main concerns in every professional effort. The less expertise required to run a process or procedure according to expected standards, the less expensive it's going to be. Therefore, we have to measure cost, length, difficulty and workforce required to accomplish the task in any proposed method or approach.

i. Complexity, skill, and difficulty: The level of the expertise required for someone to be able to find and use the meaning and lineage.
j. Duration: The time it takes to extract meaning and lineage and also to make it available to interested people.
k. Cost: Associated costs i.e. salary, training, opportunity, and etc.
l. People: The people involved in the process. This include experts who are working in the data architecture domain as well as any organizational stakeholder who would find the final results of the reporting efforts useful in order to meet their responsibilities towards the corporate mission.

3.3. Tool Supported Metadata and Lineage

The focus of this research is to provide sophisticated tool and data management support based on a metadata repository coupled with explicit data lineage mappings that will help users understand the data they are interacting with.

We attempt to provide a firm and efficient data lineage mechanism from reports to conceptual business models to data warehouse that will link report elements to the relevant metadata documentation web application that is continuously synchronized with the data warehouse. Our proposed approach is depicted in Figure **3-2**. This figure takes the enterprise data

architecture diagram illustrated in Figure **2-3** and adds the following elements which are intended to address the issues described in 3.1:

- Metadata Repository
- Lineage Mappings
- Tool support
    - Metadata Documentation Web Application
    - Dynamic Synchronization with Data Warehouse
    - Report to Documentation Linkage

The data warehouse layer embraces the raw data accumulated from different operational data sources. We propose a metadata repository that summarizes the data warehouse entities and provides descriptions about their meaning and the purpose of storing them. This element (metadata repository) provides an integrated and comprehensive environment to store all the metadata documentation and the analysis performed on them as opposed to the scattered file based documents and islands of disconnected and out of date information in current practice.

Also, Metadata Repository, as depicted in Figure **3-2** helps translate reporting requirements into the right data fields in a data warehouse. The reporting layer in which there exist dimensional data marts corresponds to a set of related reports through a fact table. We use dimensional data marts to represent cube data that generate required reports. In Figure **3-2**, elements of the proposed framework are demonstrated while showing where in the enterprise data architecture (Figure **2-3**) they can fit.

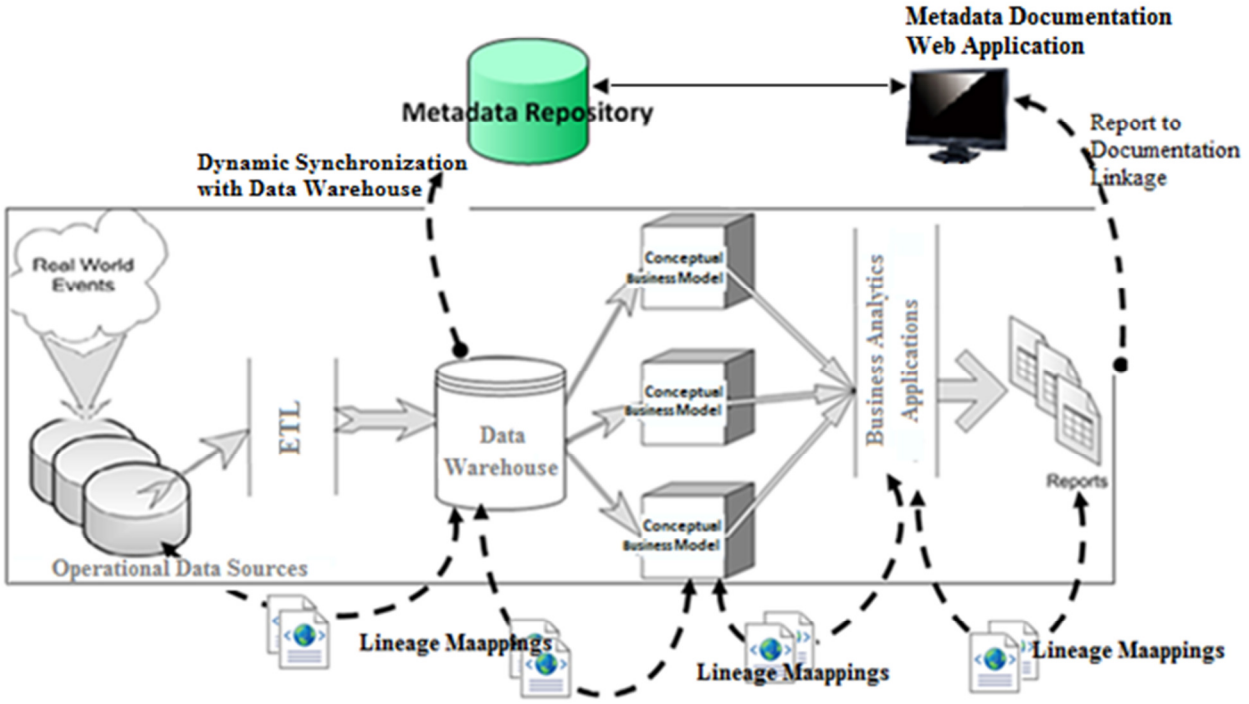

Figure 3-2: Elements of the proposed framework

The Metadata Documentation Web Application provides a user-friendly web application to view and edit the metadata info, linkage and lineage mapping and their descriptions which is shared among a wide range of users contrary to current practice, where the documentation was not reachable except for a few people.

The Lineage Mappings are recorded and documented in the web-based application to redirect users from a report element to its roots in the data warehouse and operational data sources.

Using the proposed approach, there would be no need to form ad-hoc projects for each reporting attempt. In fact the managers themselves are enabled by the tool to generate new reports (slightly different from the ones previously generated, though) and extract the meaning

and lineage of the report elements in order to fully understand the results and implications of a report and therefore make better decisions.

In addition, the data analysts, along with other stakeholders, would have documentation tools and means to communicate and share their ideas. In this scenario they wouldn't urge the other experts to abandon their agenda in order to help them create a report.

Of course our proposed approach would require some preliminary training and setup costs, but except for the starter costs and training, it wouldn't implicate the expensive experts' salary and opportunity costs. As the proposed method offers a point and click mechanism, the reporting endeavour would be wrapped up within seconds and minutes instead of weeks. In the subsequent section we describe each and every proposed element in detail.

In the long run, the proposed approach is accessible to all the stakeholders including managers, data analysts, DBAs, and domain experts. However, the DBA would be responsible for managing the mappings and the metadata repository along with their usual responsibilities for the data warehouse. Also, the head of data analysts would take the role of user administrator and all the corresponding changes to user IDs and their accessibilities.

3.4. Metadata repository

In big organizations the volume of records to be stored and the number of fields and tables in each database grows more and more each day. In order not to lose track of data and to know what pieces of data are being kept in the databases or data warehouse, we need to come up with a comprehensive addressing or referencing mechanism which shows us what's available and where each piece of data is lying. Also like any type of data, metadata requires a persistent data store in which to keep it.

As explained in chapter 2, metadata gives us the list of all the tables and columns in our data warehouse. The concept of metadata is not a new thing but what we propose is unique in its own way which employs metadata knowledge as one factor to help describe the lineage. The proposed metadata repository is a dynamic SQL database which is updated with the latest changes through the web application (as well as through dynamic synchronization with the data warehouse).

A solution to up to date documentations could be to keep track of every data manipulation and document a date stamp for any edit or change. The date stamp is the key to knowledge and info tracing. We proposed columns in the metadata repository that capture the exact time of any modification in the repository. This feature allows tracking of changes and the operators performing the change. In big organizations such as hospitals with a large number of employees, monitoring the employee's data manipulation is a necessity rather than a feature.

The metadata repository is designed to store the following data. The features mentioned subsequently are the results of a set of scheduled weekly meetings with data analysts dealing with problems and situations that motivated this thesis. Below we see a list of metadata knowledge and their intended contributions to the overall framework.

- Usual metadata

By usual metadata we mean the most popular metadata information that is a part of almost any metadata effort and repository. Studying the metadata literature and previous academic work in this area helped us pinpoint the features and include them in our proposed approach.

Over time and as the data warehouses get bigger and/or employees change, the purpose and name of an entity in data warehouse doesn't help understanding the full meaning. Therefore documentation of every entity in the data warehouse and their attributes and the availability of descriptions about them make it easier for any person to fully comprehend the data. We categorize description and caption for every table and column in the data warehouse, the latest date when a column has been updated and the info about the person who has performed the update, and categories of tables and their columns with a parent and child relationship into the usual metadata information.

The purpose of this classification and categorization was to provide a mechanism for easier navigation and searching. It is obviously easier and more time efficient to explore a class or subclass instead of the whole repository.

When it comes to maintenance and update of the old data or even applying statistical functions on the data values to measure metrics, knowing the data type and the length of the column is a vital piece of information. For example, you cannot apply discrete values functions to continuous values or inverse.

- Statistics and usage

Statistical and usage requirements were specifically recognized throughout our meetings and interactions with the local hospital professionals. Data analysts at the local hospital use the usage and statistical analysis of the values in data warehouse columns as a basis for their reports and analysis.

To provide statistical analysis of the data within the data warehouse, we require some knowledge. As such we can point at the number of attributes (columns) for each entity (table) in the data warehouse.

There are also some values that help measuring the performance indicators or metrics. Numbers of records in each column helps statisticians carry their tasks. Other statistical values such as top 5 values and their frequency in case of columns containing discrete data and maximum value in case of continuous values are also available.

- Lineage

The lineage between the data warehouse and the operational data sources has been discussed in previous academic research and also has been practiced in the local hospital under investigation. We decided to combine these data with our metadata as a matter of the utmost importance of these pieces of information and their connection to the data warehouse elements. As an example we can point at data type info in the source system as opposed to the data type of the same element in the data warehouse that shows the transformations of the data elements as they moves from one layer of the enterprise data architecture to the others.

Therefore, lineage information in metadata repository involves the source system details. The source system details in our proposed metadata repository is composed of the name of the source system feeding each column, the data type and length of the similar column in the source system database.

- Comments

During the meetings with users and documentation experts, they expressed the need to a mechanism that facilitates sharing comments and little pieces of useful experiences with data items. They needed a billboard to gather these professional hints. Therefore, we proposed commenting feature which evolved through meetings and iterations.

As a matter of having no comment sharing environment in their practice of enterprise data architecture, data stewards had a hard time finding little but vital pieces of advice from those who had dealt with similar situations. For example when a DBA or data analyst knows that the value of one column is dependent to two other columns in the data warehouse, they can add this piece of information as a comment and it would be stored in the metadata repository as some metadata. However, we should note that the quality of comments is an issue that needs to be controlled by the team leader and administrators and our proposed feature cannot certify the accuracy of a comment.

When a data steward or stakeholder looks at the data and metadata specifically, they might feel the necessity to express their ideas and comments. The common need for communications about data elements in a professional environment and lack of straight forward and pure facilities made this feature very critical.

- Attachments

Another piece of information that a data steward or stakeholder might need to keep for future references is a set of documents explaining the procedures and defining the related technical terms in the area. In current practice, a lot of these documents are stores in file based

storage units but what we aimed at was a comprehensive and integrated metadata structure which accommodates various pieces of metadata data and knowledge. The local hospital professionals asked for a storage environment which is integrated with the rest of metadata information to keep their business metadata documents.

- Web application user accounts and their encrypted credentials

In our proposed tool, we not only list the metadata knowledge but also collect specialized details and descriptions about each column in the data warehouse. The description tells us why we are keeping each specific column and what the records are going to reveal.

As we know, a data warehouse combines various databases into one integrated repository. Therefore, we need to know where each piece of data in a data warehouse is coming from or in other words we need to know their source systems. This is one piece of knowledge that the metadata repository knows about each column.

On the other hand, in the proposed documentation tool, some statistical analysis is performed on the values of each column. Also, we can gain knowledge about the data types and the length of each record. As an example of statistical analysis, we can mention the most popular values in each column and their frequency.

One of the other products of the proposed documentation tool is the means to communicate and share comments. Data stewards or other visitors of the web-pages can add comments to each column description.

This metadata repository contains technical metadata as it defines the objects that make up the data warehouse from a technical perspective. Therefore, it includes system metadata defining the data structure e.g. tables, fields, data types, indexes, etc. On the other hand, it contains business metadata as it describes the contents of the data warehouse in more user accessible terms. It tells us what data we have, where it comes from, what it means, and what its relationship to other data in the warehouse is. We even came up with some classifications and grouping of metadata objects.

In this mechanism, the DBA who was responsible for the data warehouse can periodically run a set of stored procedures to update the metadata tables in the data warehouse. This way the web-pages which are fed from all the metadata tables either residing in the data warehouse or in the metadata repository always reflect the latest data.

On the other hand, the tables which reside in the metadata repository are updated via the methods in the metadata documentation web application and the procedures stored in the metadata repository. However, the metadata documentation web application still updates the metadata tables residing in the data warehouse in cases where, for example, an authorized user changes the description of a column.

3.5. Mapping Report Elements to Metadata Repository

As we mentioned in section 3.1 Problem Definition, when a manager looks at a report, they want to know what each word and expression on a report mean, what source they are fed from, and how often are they getting updated so that the mission of a report, which is the performance measurement or the spread of knowledge, is fully satisfied. Nowadays, thanks to the various reporting applications on the market, report generation is easy but keeping the lineage

between the report and the data source is a difficult task to do, not to mention that in the process of reporting efforts, the meaning of the data elements are lost in the middle.

3.5.1 The lineage trace from a report to data warehouse

By mapping report elements to the metadata repository, we want to trace the roots of report elements back to the data warehouse and understand their meaning through the data warehouse documentation web application. But there's no direct path between report elements and the data warehouse.

The reports generated by a business analytics application are defined by a declarative report specification; an example report specification is shown in Figure 3-7 (using the Report Authoring tool Cognos Report Studio). What we suggest is to leverage the report specification that links the report generated by the BI application to the conceptual business model and then from there to the data warehouse and finally invoke the metadata documentation web application at the appropriate spot.

The reporting elements in fact are mapped to elements of a report specification. Then the report spec is linked to a conceptual business model by one to one element mapping and at the end, the elements of that conceptual business model are mapped to the data warehouse.

As mentioned in section 3.4, the metadata repository summarizes the data warehouse entities and provides a list of all the available tables and columns. Also we can access the contents of the metadata repository through a metadata documentation web application. Therefore, when the user is redirected to the data warehouse elements, they can see the descriptions and details of the related data warehouse entity on the metadata documentation web application.

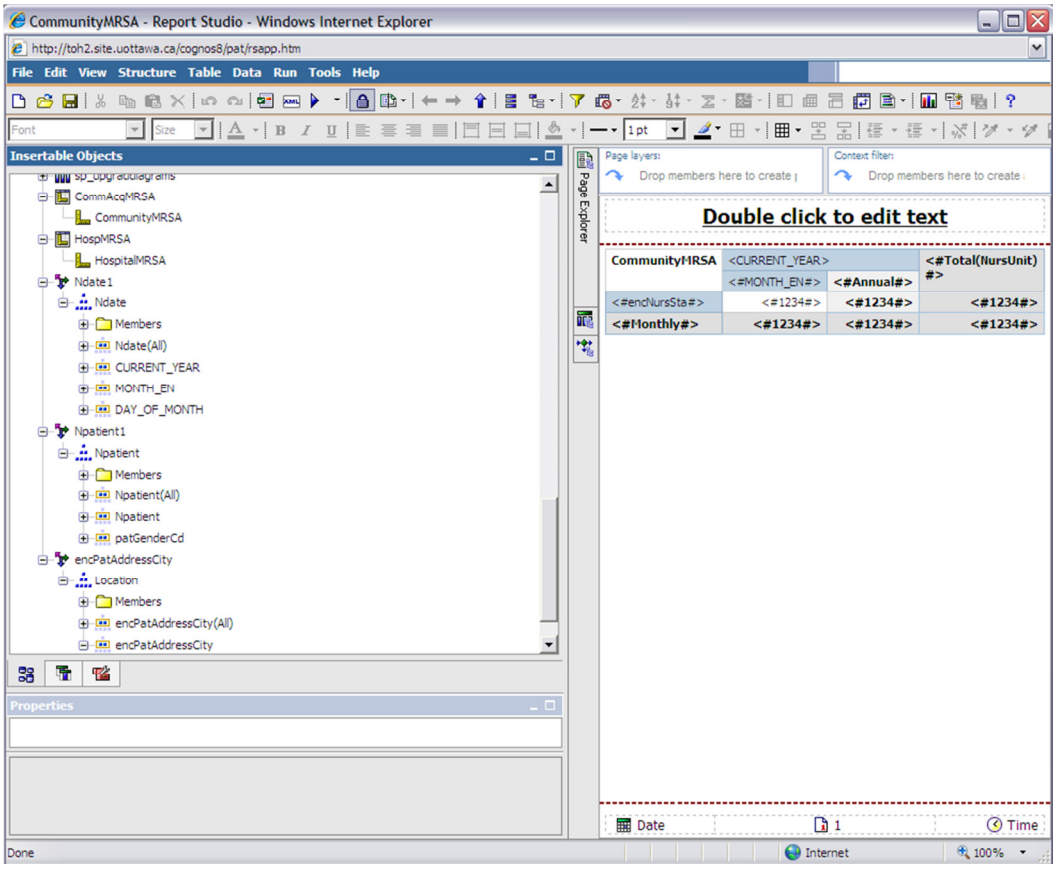

Figure 3-3: example of a report spec in Cognos Report Studio

The main idea is to provide a deeper and wider view of reporting elements that are available to end users. Reporting elements are organized into facts and dimensions in a conceptual business model. Documenting reports using a conceptual business model in which the facts and dimensions are well defined and linked, provides the semantics of the report elements as well as the relationships of its components in a broader context. This way the readability and navigation of reports is more efficient and we are able to document the items available for reporting. Furthermore, the elements of the conceptual business model are drawn from the data warehouse entities.

3.5.2 Tools to define layers of the enterprise data architecture and their mappings

Our proposed mapping structure seeks to show the links among different layers of the enterprise data architecture in an automated fashion. Therefore we need to document these links in the form of mapping correspondents and then use those mappings in a point and click mechanism to redirect us from the report to a report spec to conceptual business models and then to the metadata documentation web application that summarizes the data warehouse and how it is connected to what operational data source.

It's easier for implementers of the tool support to have the layers defined in a compatible XML format. Therefore, we have defined the conceptual business models in the CDL format (explained in section 2.5.1). Note that when using a business analytics application like Cognos, the conceptual business models can be defined using the Cognos Framework Manager tool which would be an equivalent to our CDL format of conceptual business model.

The mapping correspondents are known to be logical statements in an XML format called Mapping Definition Language (MDL) in which the one–to-one entity mappings are provided between a conceptual business model and the data warehouse entities (Nargesian 2010).

In our proposed approach, we extend the mapping correspondence to go from the report specification to conceptual business model as well. The mappings at this level are XML documents generated automatically by the business analytics application. The business analytics applications usually provide an environment to plan and sketch the report specification. These report specifications are also defined in an XML representation (Azarm, Peyton and Nargesian 2011). Typically, the report specification will reference elements of a conceptual business model by querying it.

3.5.3 Incorporation of mapping tools with enterprise data architecture

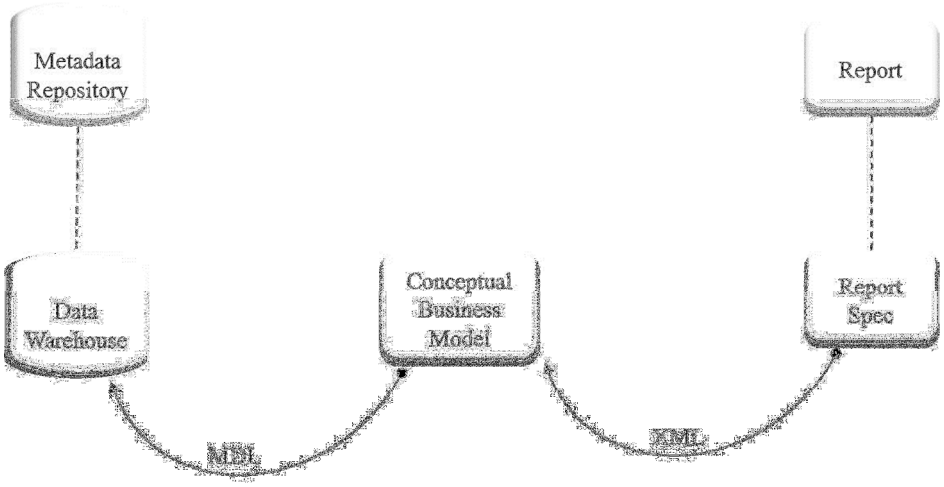

Figure 3-4: Report to data warehouse path

As shown on Figure **3-4**, we use the tools discussed in section 3.5.2 to connect the report spec to conceptual business model and then we go from the conceptual business model to the data warehouse. The XML file generated by the business analytics application connects the report spec to the conceptual business model and the MDL file defines the one-to-one element mappings between conceptual business model and the data warehouse.

Now we show the use of the mapping definition files in an example. On the report spec as presented below, we can see that the Community MRSA is an entity of the "Encounter" entity of the "Infection Control" data mart. The first mapping lineage in Figure **3-5** represents the lineage between the report spec and the conceptual business model. The second mapping is defined between the conceptual business model and the data warehouse elements. This later mapping tells us that the "Infection Control" on the conceptual business model maps to the "Nencounter" table in the data warehouse. Also the data item that we were interested in, i.e. "CommunityMRSA", maps to a column in the data warehouse called "encCommMRSA".

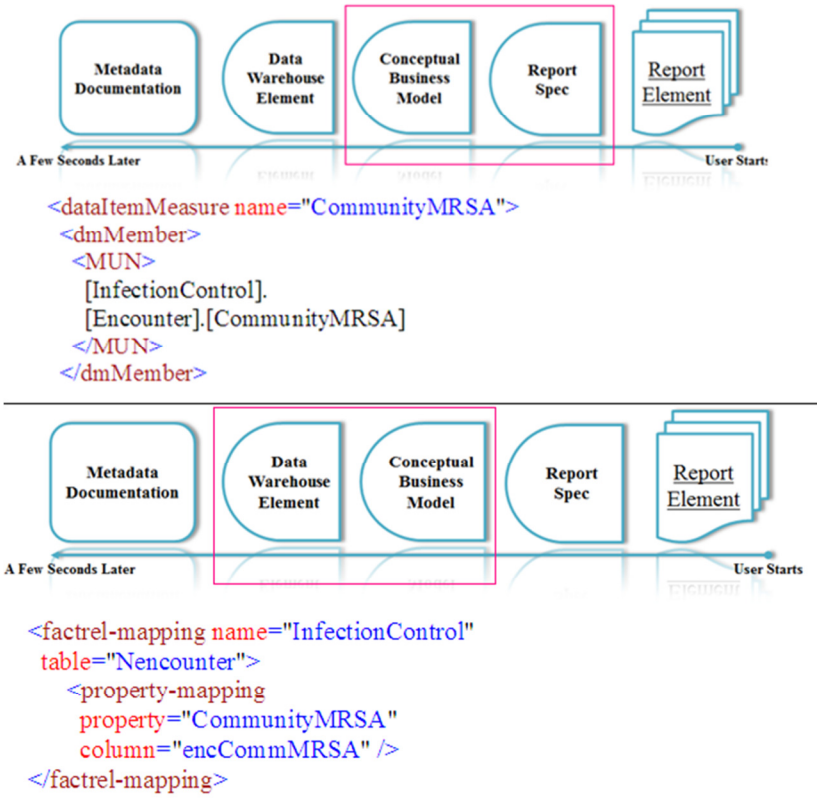

Figure 3-5: An example of report to documentation linkage path

3.6. Tool Support

3.6.1 Metadata Documentation Web Application

The importance of the Metadata Documentation Web Application is its role in enabling technology. In most organizations, there are few people who have expertise in multiple domains and know about multiple layers of the data architecture. Whereas in the proposed application, any interested person who has the authorization to view or edit certain content is empowered to extract meaning and lineage from a report all the way down to the data warehouse. Figure **3-6** presents a schematic view of the metadata documentation web application.

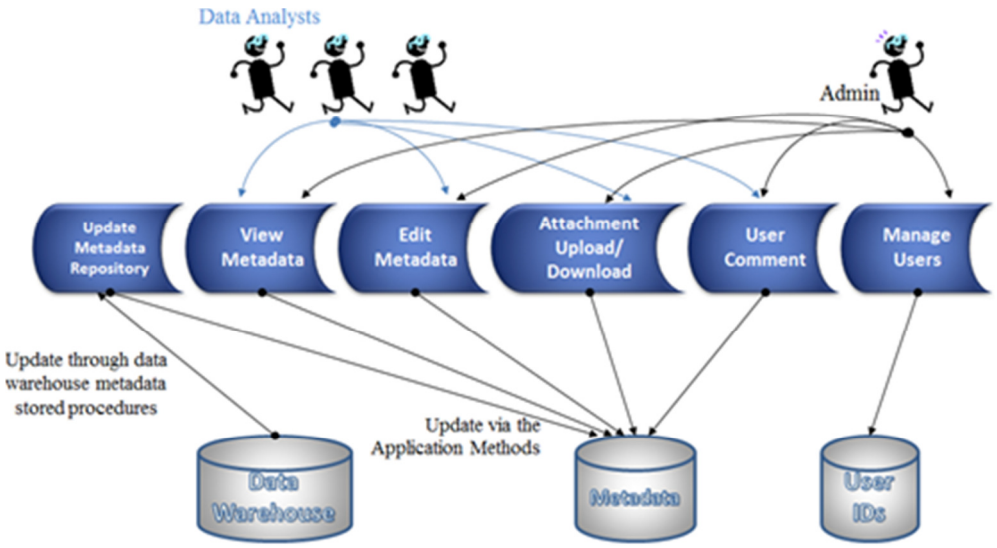

Figure 3-6: Metadata Documentation web application

The Metadata Documentation Web Application is specifically designed to help statisticians and analysts within the hospital who are in charge of preparing reports. However, the users are not limited to analysts, but anyone who is interested to know more about the data entities might use this web application to gain insight about the elements feeding the required periodic reports.

This application is the result of 5 months of iterative development including scheduled weekly interviews and reviews with stakeholders and end-users (see case study in chapter 4 for more details.

To provide for its purpose, the metadata documentation Web Application is connected to three databases i.e. data warehouse, metadata SQL repository, and SQL repository for the user IDs. However, as a matter of security issues, the data warehouse connection only allows access to carefully controlled views that provide metadata. The services the web application offers to the users are:

- To demonstrate descriptive knowledge stored in the metadata repository in charted and well organized web pages.

The proposed web application is composed of the pages (please see Figure **4-9**) that are designed with tables to accommodate a charted view of the metadata knowledge which provides sorting and editing capabilities. All the lists and tables are sortable by alphabetical order or their order in the repositories. Authorized users can edit the descriptive info, name tags, or any type of data that is not calculated by formulas and stored procedures in the application. When the update button is pressed, whatever is written in editable textboxes is saved in repositories and this way the latest changes are applied and stored instantly.

- To provide means to download and upload documents and other binary files into the metadata repository.

As explained in the metadata repository section, we needed to store binary files in the SQL repository in order to realize end-user requirements and also to have a comprehensive metadata bank. Well, now we need a mechanism for uploading and downloading those files without asking the users to gain programming and database knowledge.

The web-pages provide these facilities. Users can upload files and specify their category so after the upload is done successfully, the files are organized into the right category and dependency to the data warehouse entity.

- To provide communications platforms and means to show and store user comments.

Each time a user places a comment, a time stamp and User ID info is stapled automatically to the inserted comment. The authorized users also have the ability to delete irrelevant or old comments from the web application screen and the metadata repository.

- To establish a security and authentication mechanism in order to protect data privacy policies.

Pervasive use of an electronic and virtual environment has raised a lot of security issues. Every organization has data privacy rules and procedures to protect them. On the other hand, the data warehouse of an organization, where everything about the clients, employees and products is kept, seems quite critical and sensitive. Therefore, mechanisms to protect organizational data are of utmost importance.

In our proposed application, we impose a very strict security and authentication mechanism. User accounts and their roles are well defined and the passwords are encrypted in a data repository. The roles define which users are authorized to view, edit, and/or delete certain information.

- To enable the administrator user to modify user accounts, open new accounts, delete accounts and grant different read and/or write permissions.

As soon as a security mechanism is applied in an organization we need someone to take control over the accounts and manage them. To make the user management easier, a web-page in the application is designed for the administrator to be able to manage users in a friendly interface.

- To update metadata repository based on changes in the data warehouse

An update method in the metadata documentation application, when triggered by an event from the menu on the screen, realizes any changes in the metadata columns or tables if applicable.

This is done by a simple outer join query between metadata tables in the data warehouse and their duplicates in the metadata repository. If there were any changes, the method updates the related tables in the metadata repository. Therefore, the web application always shows the most up-to-date data.

3.6.2 Dynamic Synchronization with Data Warehouse

The system pictured up to now would be useless if it causes inconsistencies and data conflicts. The data warehouse entities are dynamic objects in which new data is entered every second, details and attributes are subject to change every once in a while, and they could be dismissed as changes occur or even new tables might be added. In order to avoid any inconsistencies with the data warehouse, we came up with our proposed synchronization solution.

Our proposed framework and toolset leverages the data warehouse. As mentioned in section 3.4, we have metadata tables in the metadata repository as well as in the data warehouse. These metadata tables are being fed by stored procedures and views specifically written to update metadata info. These stored procedures reside in the data warehouse and the metadata repository.

The dynamic synchronization mechanism with the data warehouse is summarized in Figure **3-7**. The procedures in the data warehouse include queries to:

- update the list of tables
- update the list of columns
- get the number of columns
- get the number of rows for each table
- get the data type and length of each column

The tables containing the list and descriptive info about the data warehouse tables and their columns (refer to section 3.4) are being updated by metadata procedures like "UpdateTable" and "UpdateColumn" as an example. The "UpdateTable" procedure is a metadata query that returns a list of all the tables that are kept in the data warehouse and the "UpdateColumn" procedure returns the list of their columns which then update the metadata tables.

Another piece of metadata is the number of columns and number of rows (records) each table in the data warehouse contains. As mentioned in 3.4, these numbers are very critical for the statistical analysis. Also, the data type and length of each column of the data warehouse is of utmost importance and can be found by procedures stored in the data warehouse. These stored procedures, however, need to be run by the DBA every once in a while depending on the rate of changes of data.

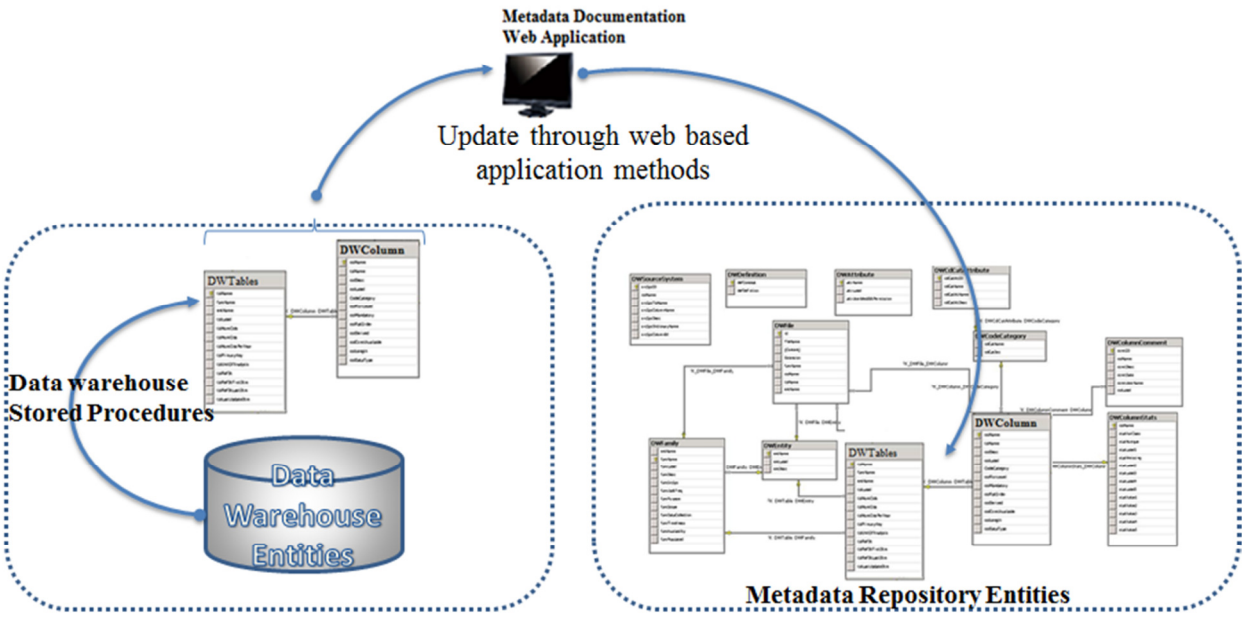

Figure 3-7: Dynamic Synchronization Mechanism

The metadata tables in the data warehouse are connected to the metadata documentation web application. When, for example, a new column is added to the data warehouse and the DBA has updated the data warehouse metadata tables by running the stored procedures, then with a single click on a menu item in the UI, the web application catches the difference and updates the list of columns in the metadata repository. As explained in section 3.6.1, this is done via a page load method in the metadata documentation web application.

Since the web-pages are connected to the metadata tables in the metadata repository in real time, updated info appears instantly on the metadata documentation web application web-pages. This mechanism makes up for our consistency and being up to date concerns. One of the conclusions of such a mechanism is that it also allows the latest analytical values because the built in functions in the application are now applied to the most up to date data.

3.6.3 Report to Documentation Linkage

In this section we review how the proposed toolset offers an automated mechanism to pull up meaning and lineage rapidly and with no technical expertise required. As we explained earlier, reports are typically generated by business analytics applications and can be customized to organizational specific conditions and requirements. Therefore, when a user is interested to know more about an element or term on the report, they simply have to click on the title of the element to view the underlying data entities and attribute(s) forming that element. Subsequently they are redirected to the metadata info on the metadata documentation web application. There they can see some description, data type and source system info, and specialized discussions about the data items related to that specific report element.

The proposed tool automates the flow of control (illustrated in Figure **3-8**) that takes a user click from a report element to the metadata documentation web-pages. The links labelled "Tool Support" show the interaction that the user experiences. When the user clicks on the title of a report element the tool takes them to a web page which shows a list of the data items from the Data Warehouse that the report element is based on. A click on each one of the data items would pull up the related pages in the metadata documentation web application where they can find all the details about the related data warehouse data item(s).

The dashed lines in Figure 3.8 show the mappings that are used to support this functionality. The mappings between the conceptual business model and the data warehouse elements describe their one to one element relationships. The application employs these parsed mapping files to pull up the right page in the metadata documentation web application and that's

where the user stops to read all the details, statistical analysis and the comments related to the underlying data items for a report.

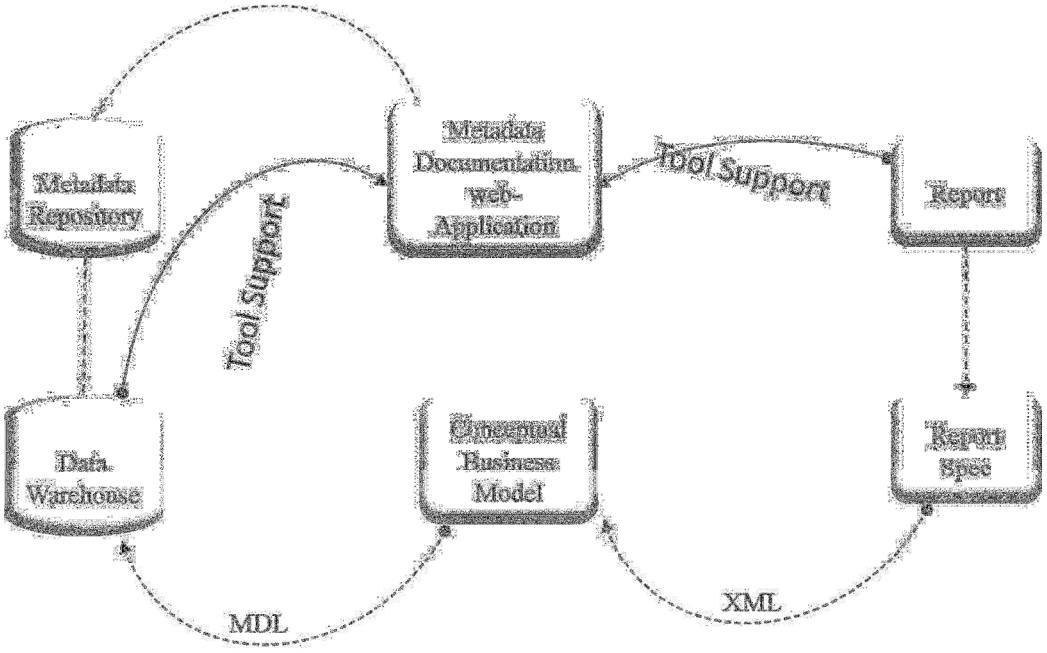

Figure 3-8: Lineage offered by the tool support

Chapter 4. Healthcare Case Study: Hospital Data Warehouse

4.1. Overview

In this section, we describe a case study we did at a local hospital in order to develop and evaluate our thesis. The local hospital followed a typical approach to business analytics applications based on enterprise data architecture as described in our literature survey in chapter 2. In particular, they experienced many of the problems and issues related to metadata documentation and data lineage that we describe in section 3.1. We give an outline of the situation in 4.2. Then we describe how we took the framework we described in chapter 3 and implemented a prototype to address the issues they were facing.

In particular, we describe our implementation of a meta-data repository and its relationship to their data warehouse in 4.3, and the development of a systematic approach to conceptual business models in 4.4. In 4.5, we describe how data lineage from reports to models to data warehouse was captured, and in 4.6 we give details of the tool support that was developed. Finally, in chapter 5 we evaluate out thesis based on the case study and the criteria we identified in chapter 3.1.

At the end of our implementation and upon the request from the local hospital, the proposed toolset has been installed on their intranet and has been in use since then. The professionals at the local hospital were eager to have a customized tool that reflects their requirements and preferences.

4.2. Current Approach to Meta Data Documentation and Data Lineage at Hospital

We focused on a local hospital to implement and examine our proposed approach. We observed some gaps and problems as a result of the complexity of their enterprise data architecture. The hospital encountered the same problems as mentioned in section 3.1.

Health-related data arrives from various related organizations such as clinics, laboratories, physician's offices, hospitals, etc. Data coming from all the aforementioned sources should be extracted, transformed, and sorted into an integrated data warehouse to provide a platform for quality of health care, data analysis, and performance assessment. Information processing from source to reports in such an organization is summarized in Figure **4-1**. Healthcare operators in different departments of the organization enter the raw data through the low level application. Usually different departments are using different software with various and sometimes incompatible data types, but at the end of the day, all those data are combined in a single data warehouse.

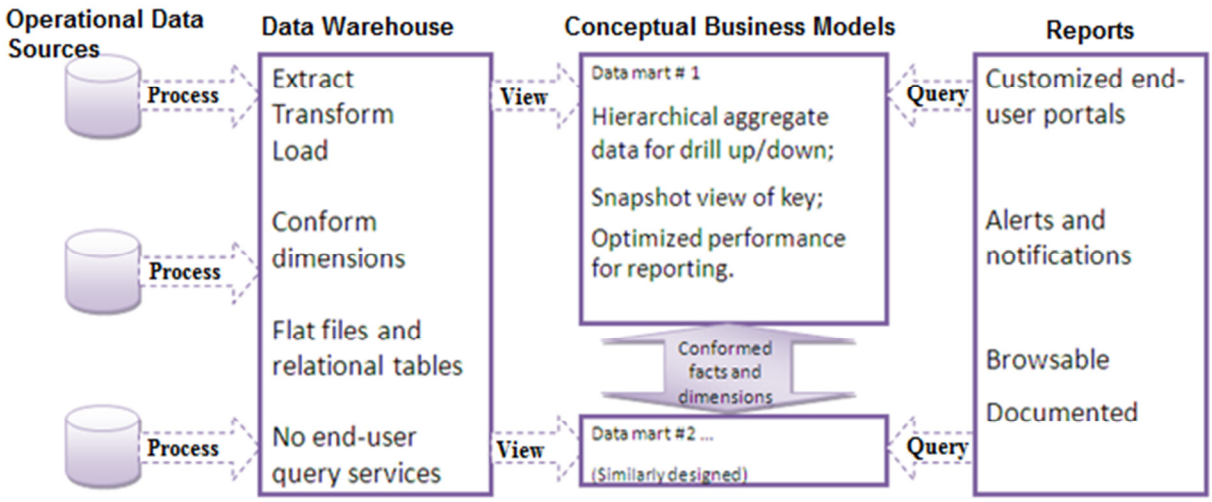

Figure 4-1: Relationship between data warehouse elements.

At the local hospital under investigation, like any other healthcare organization, the data stewards face a huge amount of data stored in the data warehouse. The reporting procedure usually starts when it's the right time for a new version of a periodic report as a performance measurement tool or when a manager specifically asks for a new report. Quite often a manager or stakeholder may ask for further documentation to understand the components and implications of a report. For example if the report showed a number for mortality rate, the managers were interested to know if the total number of patients used to calculate the rate included out patients (who did not stay overnight) or just admitted patients.

Data analysts are responsible to generate reports and the supporting documentation. They often have to refer to multiple domain experts and DBAs to help them realize and understand the right data items required for a report, their description and source information.

The reporting process would typically last days to weeks at the hospital and each time a team of domain experts, DBAs, and analysts are assigned to accomplish the job. They first had to identify the reporting elements and then realize the data items in the data warehouse that can provide for the report elements. This was the lengthy and the tricky part because of the lack of up to date documentation about the elements of the data warehouse. They had to manually search the data elements of the multiple layers from a report spec to conceptual business models and then to the data warehouse trying to extract the meaning and lineage.

The data analysts at the hospital documented their efforts and accomplishments but in static HTML web-pages, Access databases, or in scattered files. Since there was no automated system of synchronization, the documentation fell out of date quickly unless there was someone to update them manually.

The lineage between the different layers of the architecture was the missing part at the hospital, as well. The lineage between the data elements in the operational data sources and the data warehouse elements had been documented, but the rest of the architecture had been ignored.

4.3. Implemented metadata repository

In response to the weak documentation observed at the local hospital, we proposed the use of a comprehensive, integrated, and automated metadata repository. To extract reports from the data warehouse, we need to understand exactly what is stored in it. Metadata documentation can help organize and understand data within such a warehouse.

We have to point out here that during design iterations, we figured out that the tables containing the list of tables and columns of the data warehouse would be better stored inside the corporate data warehouse. Therefore, we keep one of each of those tables (two tables) both in the metadata repository and in the data warehouse.

The Implemented Metadata Repository features the following points:

- Categories of tables and their columns, description and caption for every table and column in the data warehouse

During our periodic meetings with the hospital we developed name tags and descriptions for the data warehouse entities. Furthermore, the entities were classified in a hierarchical set of categories. We introduced an “Entity” group as the highest level category. The entity group provides a filter that separates entities based on their functional purpose and contribution to healthcare services. Each “Entity” is composed of a few “Families” and then each “Family” entails some “Tables”.

Family categories provide for the second layer of filtration and combines entities within the same subdivision at the hospital. The “Table” category accommodates all the tables in the data warehouse. At the end of the structure fall all the columns within the data warehouse. In the case of healthcare data, each “Table” embraces a long list of columns.

The aforementioned structure builds the schema of the proposed metadata repository. This repository has been developed in a MS SQL Server database engine. In Figure **4-2** we can see the complete schema of the proposed metadata repository for the healthcare sector.

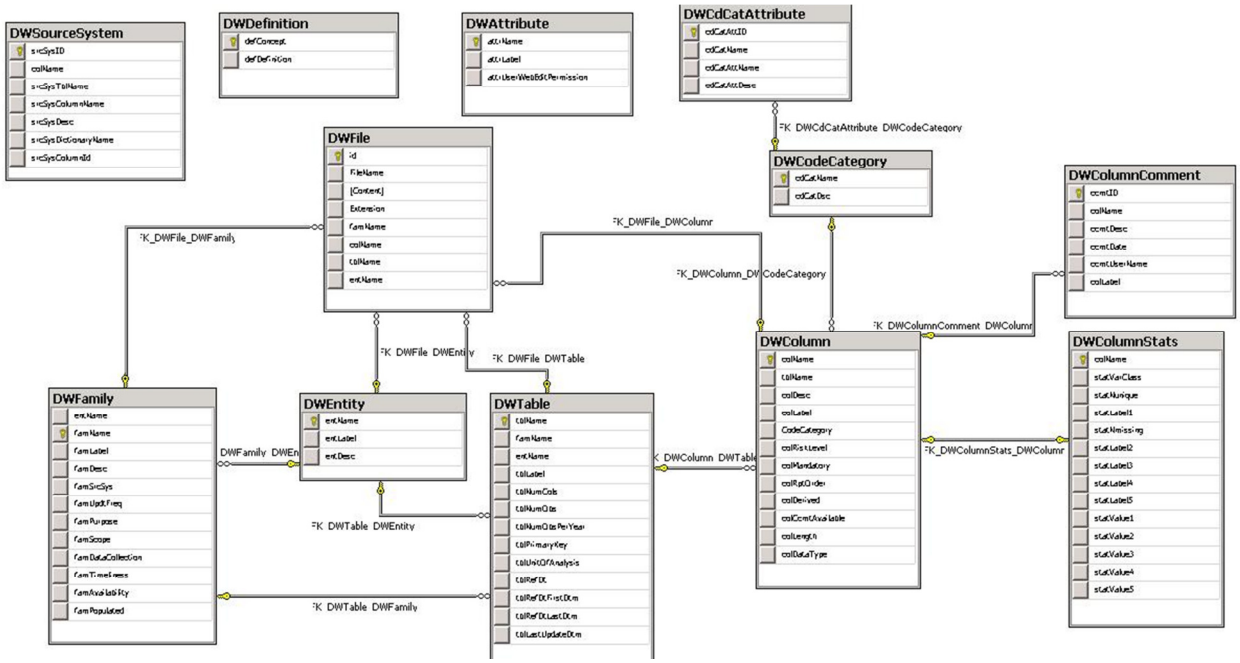

Figure 4-2: Metadata Repository SQL Schema

- Number of columns of each table

To help generate further statistical analysis, a stored procedure in the metadata repository is dedicated to calculating the number of columns in each table. This is one step towards keeping up to date with the latest changes in data.

- Number of records (instances) stored in each table annually and in total

As described in chapter 3, we generate some views and procedures to measure the performance indicators or metrics. We can see the SQL code subsequently in section 4.6.2. This value might not be seen directly but indirectly it contributes to the calculation of other statistical values. For example if we want to measure the rate of Methicillin resistant patient at hospital, we need to know the number of all the patients in a specific time span.

- Source system knowledge

Source system knowledge is the key to our proposed system of lineage with the source data. This piece of information in our framework is entered manually by the DBA each time they are adding a new entity to the data warehouse and is reflected in the metadata repository as well.

- Data type and length of the column

This is another piece helping the analysis and metrics measurements. To know which statistical functions to apply to each piece of data, we need to know their type i.e. either discrete or continuous and their structure which can be integer, character, and etc. For example, the encCommMRSA column in the data warehouse contains discrete values stored in an integer data structure.

- The latest date when a column has been updated and the info about the person who has performed the update

The date stamp is carried out through a stored procedure. This feature addresses a critical requirement from the data analysts to keep track of the changes. This feature is incorporated among the web application features and the results are stored in the metadata repository. For example, when a data analyst edits the description of a data element, the exact time of this edit and user ID of the analyst are stored in the metadata repository.

- Statistical analysis

More statistical values are provided by stored procedures. Unlike the previous analytical data which were related to warehouse entities, these values provide insight about the records and transactions forming the healthcare services offered to customers. For example, to know the most popular symptom in patients who took the chemotherapy, we need to find the most popular value in the column in the data warehouse that stores the patient symptoms with the condition that the patient type is cancer.

Other information provided through this feature are top 5 values and their frequencies, minimum and maximum values, the highest and lowest values, number of missing values, and number of unique values The procedure below as an example generates such info automatically for all the columns inside the data warehouse.

```
Create view [dbo].[ViewNumOfMissingValues] as
Select
        substring(col.name, 1, 50)  as column_Name,
        sum(ind.rows) as Number_of_Rows
From
        syscolumns as col
Inner join
        sysindexes as ind
        on col.id = ind.id
Where
        col.name <> null
GROUP BY col.name
```

```
Create view [dbo].[ViewNumOfMissingValues] as
Select
        substring(col.name, 1, 50)  as column_Name,
        sum(ind.rows) as Number_of_Rows
From
        syscolumns as col
Inner join
        sysindexes as ind
        on col.id = ind.id
Where col.name <> null
GROUP BY col.name
```

- Comments about each column of the table made by web application users

This feature corresponds to the data stewards' requirements towards providing an environment for specialized comment sharing in the most convenient place. Comments posted on the pages of the proposed web application are stored in the metadata repository since they are meant to be related to metadata entities and state some metadata ideas. The commenting feature is discussed in the tool support section.

- Usability guides and documents related to each category.

In the proposed metadata repository we dedicated a table to store binary files that the data stewards needed to review frequently. This is in accordance with having a comprehensive unit

for storing all the metadata related material. These files include specifically those former scattered descriptive documents in the binary format (.doc or .pdf) generated by data stewards in their former projects. Now, we have dedicated a spot in our integrated metadata repository to store them.

- Web application user accounts and their encrypted credentials

In the proposed tool support, secured web-pages are provided. The credentials and user roles and authentication methods are stored in a SQL server database. For example in the local hospital we created three roles for the metadata documentation web application i.e. Analysts, public, and admin. These roles are stored in the User IDs repository along with the usernames, their passwords, and user's contact info.

4.4. Conceptual Model of Infection Control Report

To familiarize ourselves with the common procedures and terms in a health care environment, a brief explanation for admission procedure at the local hospital under investigation is presented below:

As a patient refers to an information desk at a hospital, a new record is added to the "Encounter" entity residing in the data warehouse (see Figure 4-3 below). There could be several different types of encounters with different problems. Each encounter is connected to one patient. However, one patient might have multiple encounters due to various visits. One way of classifying the patients is to differentiate patients with different entry and exit dates. Analysts may keep track of patients in different time slots for signing in and out.

One of the data items stored about each encounter is to identify whether or not they are resistant to Methicillin. This situation can happen in any environment but what matters to the

hospital is whether they caused it. Therefore for every record in the Encounters table of the data warehouse, we can see two pieces of data called Community-Acquired MRSA and Hospital-Acquired MRSA which can only take 0 or 1. 0 means negative MRSA status and 1 indicates a positive status.

As we mentioned earlier, we use dimensional data-marts to help generate reports. The FM model [or Conceptual Visual Model (CVL)] demonstrated in Figure **4-1** is our implementation of a data-mart. The first step in creating a data-mart is to define the right fact table and dimensions. To do so, we refer to our Metadata repository. Consulting the metadata repository using the web-based tool, we understand the data elements in a performance measurement document or a report, like our case: Infection Control.

In a conceptual viewpoint, a report reflects a fact table which is the kernel part of a data mart (dimensional cube). In the process of identifying the dimensional elements, we also review the healthcare procedures related to our Infection Control example.

In this case, we associate the central fact table with the main issue being measured. Therefore, the fact table is going to be "Infection Control" as we are measuring the infection control. Using the standard dimensions in healthcare and performing customized analysis on data requirement of the hospital, and in regards to our level of data abstraction, we selected the following dimensions: encounter, patient, services, date, and location.

Encounter dimension bears related attributes coming from the Entity table in the data warehouse. For example, on the encounter dimension we can drill through encounter type or admission type.

Patient dimension on the other hand comprises elements related to patient entities. One of the hierarchies in this dimension is gender. Using this hierarchy, we can narrow down the results

of a report to a single gender only. Services dimension concentrates on different departments of the hospital like laboratory or radiology.

Almost any dimensional design has a time dimension since it's a critical element to most reporting efforts. Most of the managers are interested to assess the metrics and measure the performance in different time slots. Like most dimensional models, our time dimension has a hierarchy of year, quarter, month, and day. Figure **4-3** shows our conceptual business model for Infection Control reports.

And finally, location is a dimension with nursing station, campus, city, province, and country divisions and hierarchies. Any location-related reporting element can be drilled through this dimension. One of these elements is the encounter location.

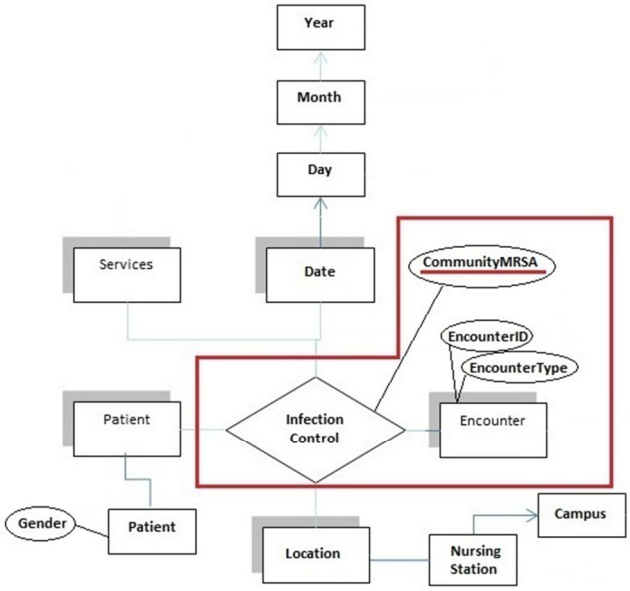

Figure 4-3: Our conceptual business model implemented for our healthcare case.

The Conceptual Definition Language (CDL) description of the conceptual business model depicted in Figure **4-3** is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<CDL>
<Schema Namespace="InfectionControl" xmlns="http://site.uottawa.ca/cim">
<!--Dimension Levels-->
        <levelset>
            <level name="Encounter">
                        <key name="encID" type="String" nullable="false" />
                        <property name="name" type="String" nullable="true" />

        </level>
            <level name="EncounterType">
            <key name="encType" type="String" nullable="false" />
        </level>
        <!--Levels-->
        </levelset>
    <!--Hierarchies-->
     <hierarchySet>
       <hierarchy name="EncounterTypeH" bottomLevel="Encounter">
            <parent-child parent="Encounter" child="EncounterType" cardinality="1..*" />
       </hierarchy>
       <!--Hierarchies-->
    </hierarchySet>
    <!--Dimensions-->
     <dimensionSet>
       <dimension name="EncounterType">
         <hierarchy name="EncounterTypeH" />
       </dimension>
              <!--Dimensions-->
     </dimensionSet>

    <!--factRelationships-->
     <factRelationshipSet>
       <factRelationship name="InfectionControl">
         <role name="Encounter" />
         <!--Rolls-->
            <measure name="HospitalMRSA" type="int32" nullable="true" />
            <measure name="CommunityMRSA" type="int32" nullable="true" />
                        <!--Measures-->
       </factRelationship>
     </factRelationshipSet>
  </Schema>
</CDL>
```

4.5. Example Report and Data Lineage

4.5.1 Infection Control Report

We chose a typical, representative report to show how the lineage is achieved through our proposed mappings. We start from the report a manager can see on a screen and we illustrate how by using our data lineage framework, the manager gains insight about the semantics of the

report elements, their roots to the data warehouse and elements available for further reporting needs.

In our example case, we used Cognos Framework Manager to build the conceptual business model and using Cognos Report Studio, we generate the infection control report from the Framework Manager (FM) conceptual business model. Figure **4-4** shows a screen shot of the infection control report.

Cognos Viewer

CommunityMRSA	2,003					2,004								2,007			Total (NursUnit)
	January	August	July	November	**Annual**	June	April	May	August	March	July	February	**Annual**	August	October	**Annual**	
Arthrities	2				**2**	22							**22**				**24**
Birthing							16	14					**30**				**30**
Cardiology	8	1	1	1	**11**	0	0		18	26			**44**	13	23	**36**	**91**
Dental						13					19		**32**				**32**
Dermatology	12				**12**				23				**23**				**35**
Paediatrics												22	**22**				**22**
Monthly	**22**	**1**	**1**	**1**	**25**	**35**	**16**	**14**	**41**	**26**	**19**	**22**	**173**	**13**	**23**	**36**	**234**

Figure 4-4: Screen shot of example report in Cognos Report Studio

As seen in Figure **4-4**, we are measuring Community-Acquired MRSA rates in different nursing units over different months. But when the manager looks at the report, they might wonder, for example, what Community MRSA means and where it's coming from. As seen on the figure, there's no explanatory data about the elements of the report like "CommunityMRSA" (circled).

In our example health care case, we generate “Hospital and Community acquired MRSA” as a periodic report. One of the key data elements on that report is “Community acquired MRSA” which is calculated through the following formula:

$$\text{(Summation of community acquired MRSA)} = \sum(encCommMRSA = 1)$$

The report is defined by an XML specification shown below (created in Cognos Report Studio). In the report studio XML file, community-acquired MRSA can be spotted in the “data item measure” tag.

Report Studio XML:

```
<queries>
  <query name="Query1">
  ...
    <dataItemMeasure name="CommunityMRSA">
      <dmMember>
        <MUN>
          [InfectionControl].
          [Encounter].[CommunityMRSA]
        </MUN>
      </dmMember>
      <dmDimension>
        <DUN>
          [InfectionControl].[Encounter]
        </DUN>
      </dmDimension>
    </dataItemMeasure>
  </query>
</queries>
<layouts>
  <layout>
    <reportPages>
      ...
      <crosstab refQuery="Query1"
        horizontalPagination="true"
        name="Crosstab1">
          <crosstabRows/>
          <defaultMeasure
            refDataItem="CommunityMRSA"/>
      </crosstab>
    </reportPages>
  </layout>
</layouts>
```

According to the XML file, CommunityMRSA is a data item measure that comes from the Encounter entity and dimension on the conceptual business model which is connected to the Infection Control fact. This relationship is circled in red in the conceptual model shown in Figure 4-3.

4.5.2 Data Lineage through MDL

To capture data lineage, we used Mapping Definition Language (MDL). An MDL associates the conceptual model (FM conceptual business model) to the data warehouse. Simple attribute-to-attribute mappings are defined between the columns in data warehouse tables and attributes in the conceptual model.

Then, a mapping compilation algorithm combines the info and generates a view (mapping) for each level, dimension and fact relationship in the conceptual model (Nargesian 2010). This way, for each level or fact in the conceptual model or report we can generate a SQL query in terms of data warehouse tables, which can illuminate how the data fields in reports are fed.

Data lineage provides insight about the elements of a report and the extracted knowledge. In our Infection Control report, the elements such as fact table, dimensions and hierarchies are documented within mapping languages which explain one–to-one mapping between report elements and data warehouse columns. Note that such mappings have a generic context that can be applied to any reporting endeavour. As a manager or stakeholder reviews these documents, they would understand each building block of the report.

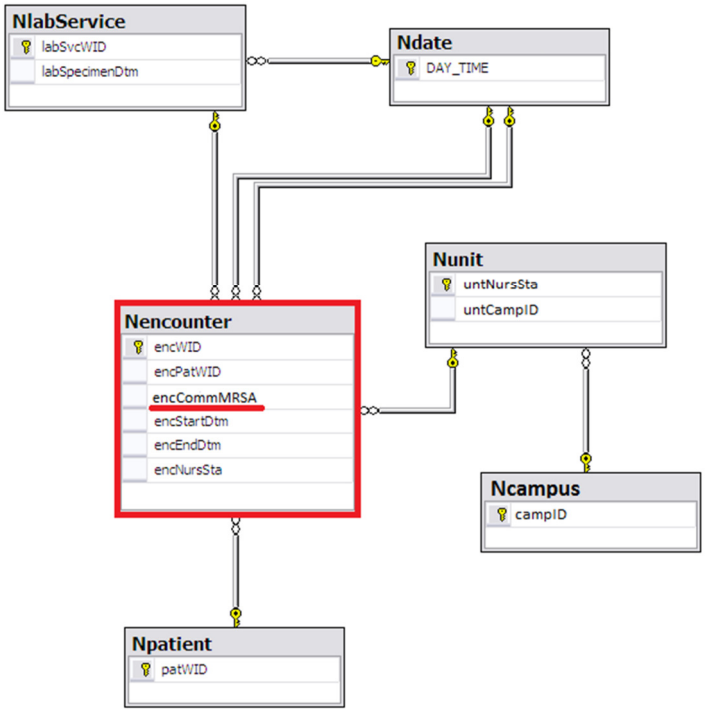

Figure 4-5: Infection control database schema

Figure **4-5** shows parts of our example data warehouse schema. As you can see in Figure **4-5** and according to MDL, "encCommMRSA" is a column in the "Nencounter" table which bears the info for Community Acquired MRSA rates.

Reports are created based on data marts, and documented using the related conceptual business model. The mappings, defined in MDL, provide clues on how the data in data marts and reports are related to the tables of underlying data warehouses. Infection Control MDL is:

```
<MDL>
  <schema namespace="InfectionControlMapping"
 xmlns="http://site.uottawa.ca/cim">
       <level-mapping name="Encounter"
         Table="Nencounter">
              <property-mapping property="EncounterID"
                column="encWID" />
              <property-mapping
                property="EncounterType"
                column="encType"/>
       </level-mapping>
       <factrel-mapping name="InfectionControl"
         table="Nencounter">
              <property-mapping
                property="CommunityMRSA"
                column="encCommMRSA" />
       </factrel-mapping>
  </schema>
</MDL>
```

As we can see in the MDL file, CommunityMRSA is the report property that maps to the "encCommMRSA" column within the "Nencounter" table in the DB schema and also a report called "Infection Control".

4.6. Tool Support

In the following sections, we go through our tool support realization for healthcare data.

4.6.1 Infection Control Metadata Documentation Web Application

In order to access, modify and manage metadata, a web-based tool was built to provide full access to the metadata repository and limited access to the data warehouse for all stakeholders. The features of the documentation tools have been completed after a series of interviews with data analysts and DBAs at the hospital. Their requirements and preferences were collected and translated into this web-based application through multiple iterations while its practicality was tested by the final users. To help manage the iterations and requirements, software development tools were used, including the Assembla® hosted website which helped accelerate software development by providing ticketing and issue management, subversion control, wiki, and etc.

The proposed tool is based on the metadata repository explained in section 4.3, and accountable for all the steps in the data delivery architecture. In fact, this web-based tool is developed to access and manage that repository as well as providing a nice UI for presenting the documentation. It provides up to date documentation in a user friendly environment. Also the automatic synchronization mechanism built into the application helps address some of the problems explained in section 4.2. The metadata documentation web application provides the following features for the local hospital:

- To provide details about the column and what it's intended to capture.

The application manifests categorization and grouping of related data warehouse entities. The application summarizes the warehouse entity titles as a left hand side menu. A click on each entity would show further details about the entity, family, table, or the attributes of a table. The descriptions and name tags for each column in the data warehouse are defined and gathered in the specified metadata tables.

- To provide info about the source system injecting the records

The source system info is reflected on the tables on web-pages as an item on the left hand side menu is clicked (refer to Figure **4-7**). The name of the source system, source system data element name, source system dictionary name, and source system description is provided about each column of the data warehouse.

- To enable data analysts to edit some descriptive info about a column

One of the main functionalities of the web-based documentation application is the editing functionality that is only open to certain users. Authorized users which are usually the data analysts and statisticians are able to change nametags and descriptions, provide more details about some elements, change the risk level of an attribute, change the naming system, indicating the primary key of a table or change it and change the reference date to an attribute.

- To share downloadable files and documents

The web application provides the facility for uploading files while mentioning which category of data warehouse entities they belong to. Then the uploaded files are shown in a table and ordered based on the families they belong to. By clicking on each file title, users can download and save them in their file system.

- To calculate some statistical variables over the values of each column

For example, minimum value, median value, top 10 values, etc. related to tuples of each column. The statistical analysis helps managers realize their critical and sensitive areas and the need for capturing periodic reports about them. In Figure **4-9**, we can see full statistical information provided on the web-pages and how the application presents them.

- To provide a basis for communication for all stakeholders

Through the web application, users can share their comments about each column. Each time a comment is posted, the table of the comments is updated with the comment text as well as the username of the person who posted the comment and the date of the post. Certain users who are granted special authority are able to delete unwanted comments from the web-page and the metadata repository. The commenting feature is specifically useful to data analysts and system developers to share their experience and/or knowledge.

Figure **4-7**, shows a screen shot of a page in our web-based metadata documentation tool. Looking at the web application we can understand the schema and structure of the warehouse and we can observe what is stored in each column of the existing tables.

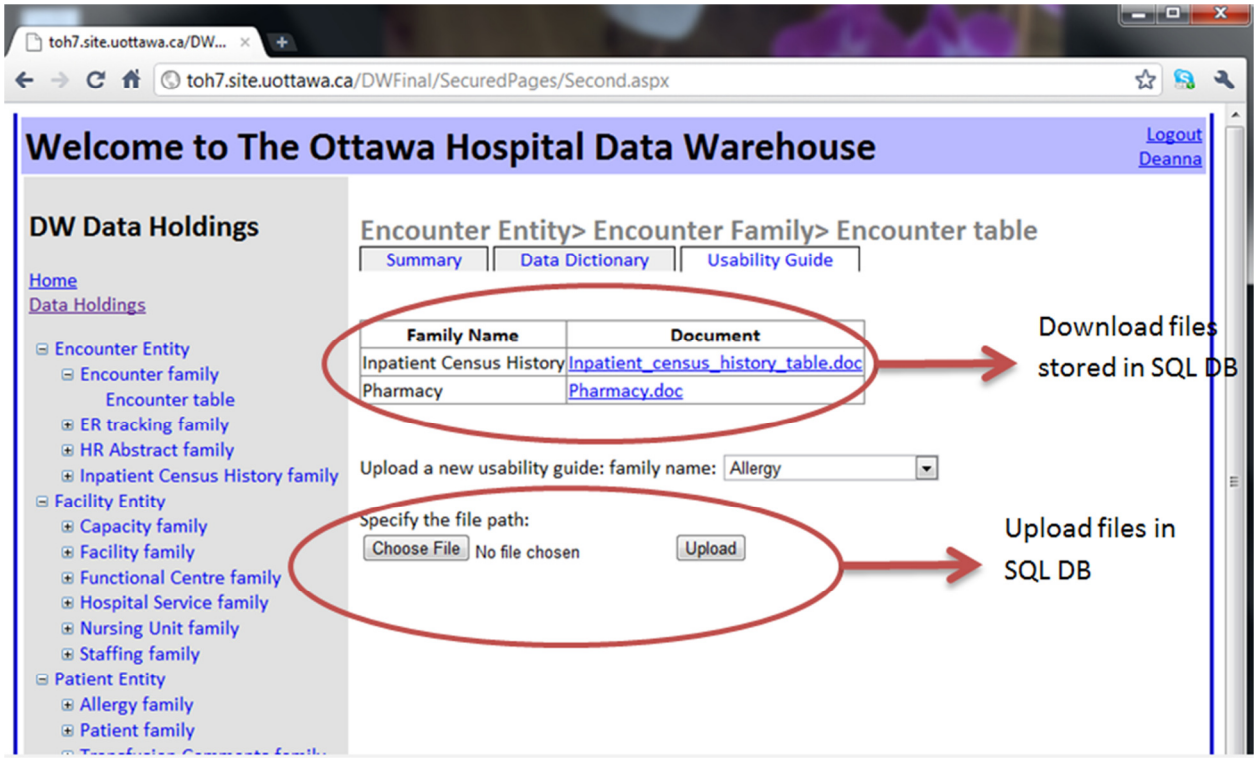

Figure 4-6: Upload and download capabilities

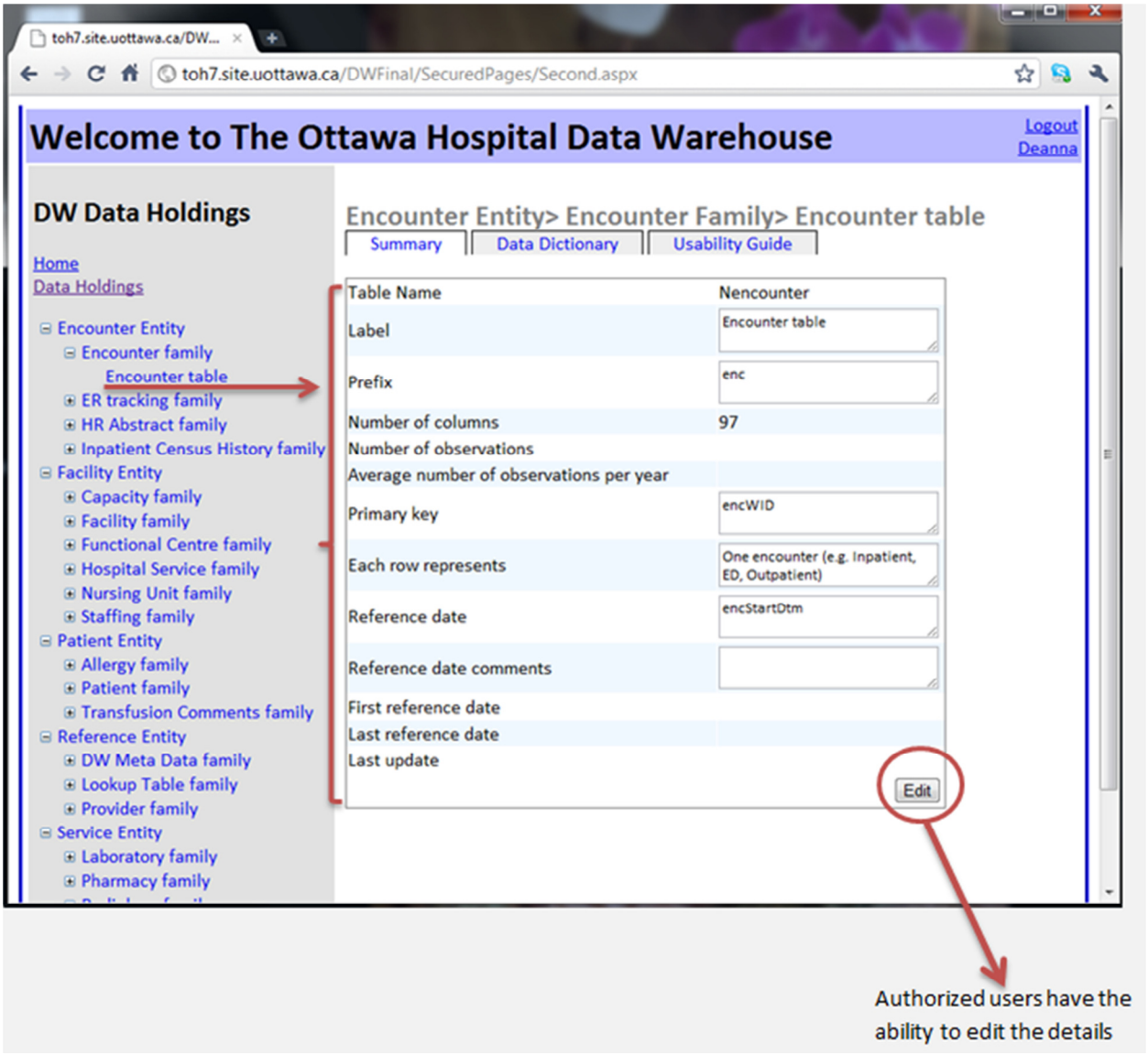

Figure 4-7: Table level details provided on web-pages

As seen on the screen shot, the details about the encCommMRSA column are:

Table 4-1: Example column details available on web-pages

Description and details	
Column Name	encCommMRSA
Description	Community Acquired Methicillin-resistant Staphylococcus Aureus
Details	Methicillin-resistant Staphylococcus Aureus (MRSA) is a bacterial infection that is highly resistant to some antibiotics.
Length	4
Data Type	Integer
Risk Level	Low
Source System Info	
Source System (SS)	SMSEncounters
SS Data Element Name	CummunityMRSA
SS Dictionary Name	C0830 (PT TYPE)
SS Description	SMS Encounters
Statistical Analysis	
Number of unique values	0
Number of missing values	60
Variable Class	Discrete
Minimum value	0
Maximum value	1
Top 1 value	1
Top 1 frequency	1
Top 2 value	0
Top 2 frequency	0

Application Design Iterations and Issues

When the application development process was just starting, the main issue at hand was to recognize the right set of requirements to fully meet the users' needs and improve their working conditions. We used ticketing and version control online tools[1] to improve the productivity and efficiency of our software development process. Therefore, the requirements were documented, prioritized, and closed once accomplished.

[1] http://www.assembla.com/wiki/show/mana-datawarehouse

At the final stages of development, we encountered the so-called integration problem. We had to provide the most up to date information to keep our users interested. In the beginning, the metadata documentation was meant to connect to the metadata repository. However, the metadata repository itself could go out of date easily.

The only unit in the data architecture which was being updated manually or automatically from the source systems with the latest changes should have been the data warehouse and there should have been mechanisms taking care of adapting the metadata repository and web-pages to the inevitable changes.

Therefore, we moved some primary metadata tables to the hospital data warehouse and developed some methods to update the rest of the tables residing in the SQL metadata repository.

4.6.2 Dynamic Synchronization with Data Warehouse

As mentioned in section 3.6.2, the synchronization with data warehouse is achieved through a set of stored procedures and also the methods built in the metadata documentation web application. Subsequently, the SQL queries for the aforementioned stored procedures are presented. In contrast to normal SQL queries, the queries below deal with metadata tags.

- update the list of tables and columns

```
Create procedure [dbo].[ListTblNumCols] As
SELECT     sys.sysobjects.name AS tblName, sys.syscolumns.name AS colName
FROM    sys.sysobjects
INNER JOIN sys.syscolumns
ON sys.sysobjects.id = sys.syscolumns.id
```

- get the number of columns for each table

```
Create procedure [dbo].[ListTblNumCols] As
SELECT     sys.sysobjects.name AS tblName, Count (sys.syscolumns.name) AS numCol
FROM    sys.sysobjects
INNER JOIN sys.syscolumns
ON sys.sysobjects.id = sys.syscolumns.id
Group by sys.sysobjects.name
```

- get the number of rows for each table

```
Create view [dbo].[ListTblRowCount] as
SELECT
   sysobjects.Name, sysindexes.Rows
FROM
        sysobjects
INNER JOIN sysindexes
ON sysobjects.id = sysindexes.id
WHERE
        type = 'U' AND sysindexes.IndId < 2
GO
```

- Get the data type and length of each column

```
Create view [dbo].[listOfColumns] as
SELECT   SysObjects.[Name] as TableName,
        SysColumns.[Name] as ColumnName,
        SysTypes.[Name] As DataType,
        SysColumns.[Length] As Length
FROM
        SysObjects
INNER JOIN SysColumns
ON SysObjects.[Id] = SysColumns.[Id]
INNER JOIN SysTypes
ON SysTypes.[xtype] = SysColumns.[xtype]
WHERE
        SysObjects.[type] = 'U'
GO
```

4.6.3 Report to Documentation Linkage

A stakeholder who is viewing a managerial report should be able to understand the report in terms of the root data elements residing in a data warehouse. Our intent was to implement a tool as described in section 3.6.3 as a fully automated click and redirection from the highest level aspect of the report to the lowest level element in the data warehouse.

However, after deploying Cognos as the business analytics application for reports, we found it remarkably complicated to merge Cognos Report Studio specifications and Cognos Framework Manager conceptual business models with our CDL, MDL and SDL notations.

We could automate the lineage path from the source system to the data warehouse and then to the conceptual business model for a report. But when it comes to the generated report by IBM Cognos with its own XML grammar which differs from our CDL, SDL, and MDLs, it was difficult to resolve differences between Cognos's notation and ours. Therefore we did not implement a full solution as proposed in section 3.6.3. However, we eventually developed a quick and dirty text search function in the mapping files to show the potential of the proposed method and left the fully automated point and click mechanism for future work. The way this search works is as follows:

When the managers need more info about a report element on a business analytics application report, they can type the name of that report element (or a business term comprising a substring of it) in the report documentation web application. Then, the application searches the CDL file which describes the conceptual business model, the SDL file which is a definition file similar to CDL except that it describes the data warehouse elements in a mark-up format, and the MDL file which describes the mappings. If there exists any data warehouse data items in the

search results, they are shown as hyperlinks which redirect the user to the metadata documentation web-pages for further details. For example, if "MRSA" is typed in the search bar, the results would be what Figure **4-8** illustrates. By clicking on a data warehouse element which is shown as a hyperlink, say encCommMRSA, we are redirected to the documentation about the encCommMRSA column in the metadata documentation web application as shown in Figure **4-9**.

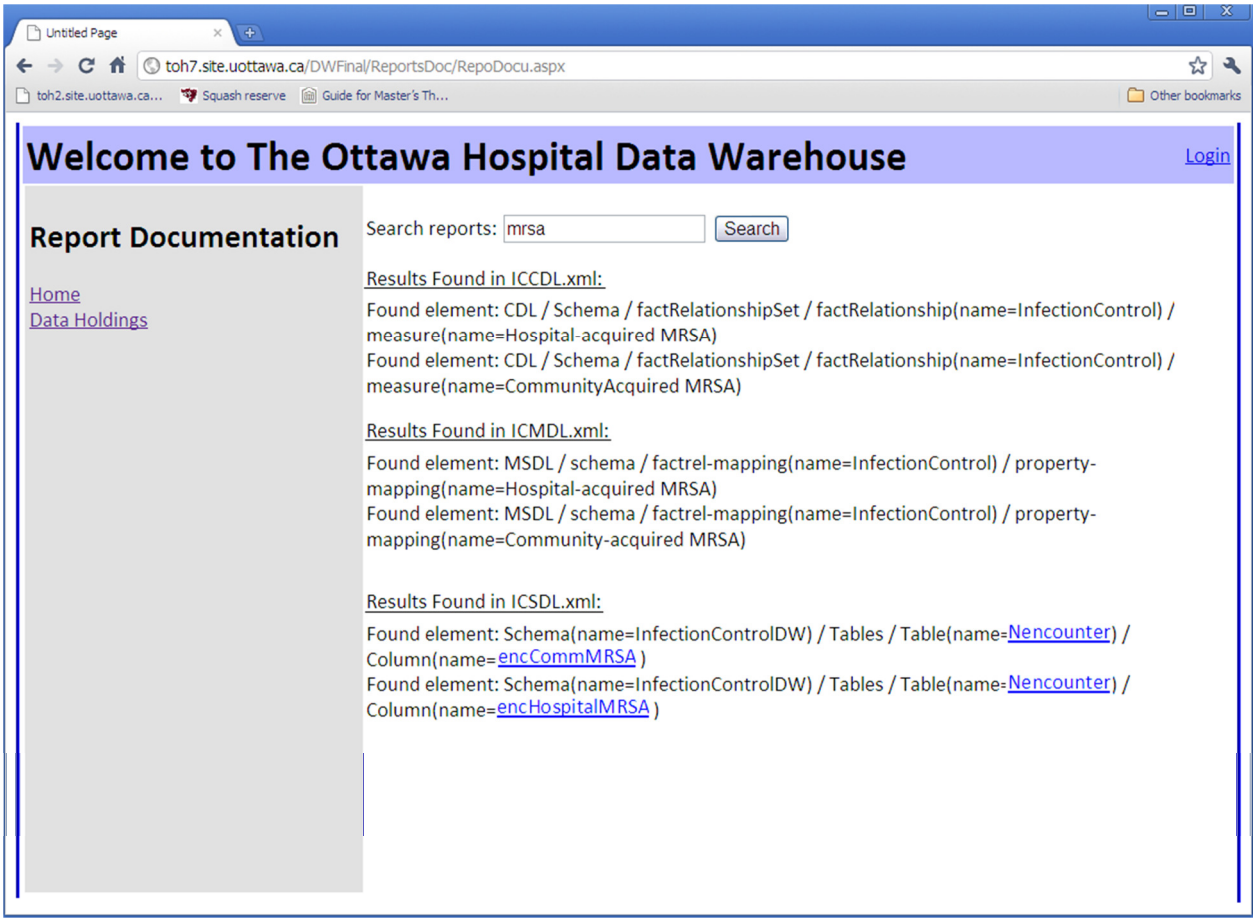

Figure 4-8: Text search to retrieve lineage

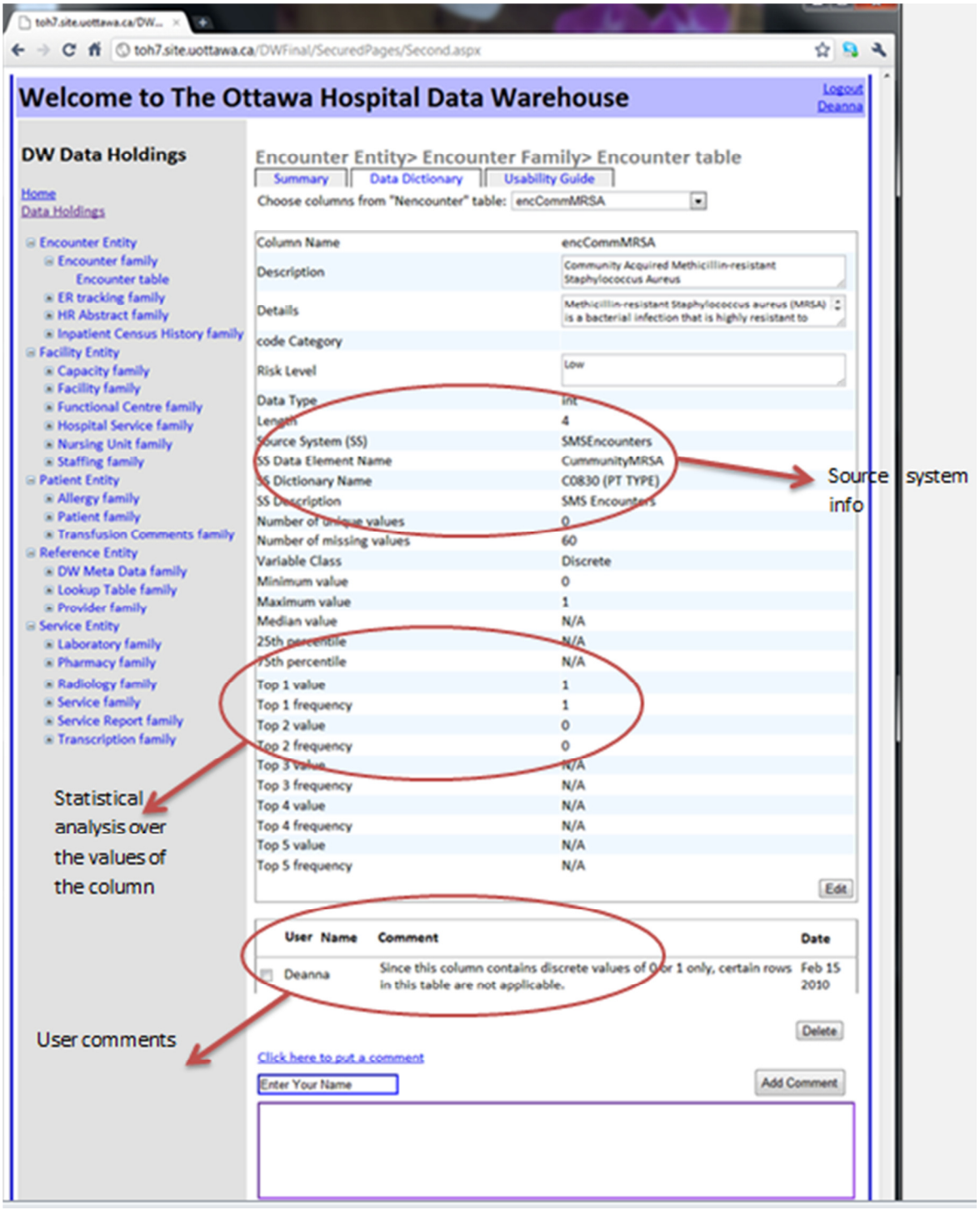

Figure 4-9: encCommMRSA Documentation on the proposed application

Chapter 5. Evaluation

In this chapter, we evaluate our proposed framework for Tool Supported Metadata and Lineage as a means of addressing the documentation criteria that were identified in section 3.2. We compare our framework against the Current Enterprise Data Architecture Approach that we described in chapter 2, using the case study we performed at a local hospital as described in chapter 4. The hospital approach was typical of current enterprise data architecture approaches, and we prototype our framework as a means of addressing some of the issues they were facing. In particular, we compare the two approaches along the three major themes of Metadata documentation, Availability of Documentation and Documentation Effort.

Note that our research methodology is design-oriented research as described in section 1.4, we are not attempting a complete empirical evaluation our framework is complete and correct, but rather we are simply trying to demonstrate the utility of our approach in addressing the gaps we have identified. Future work will need to develop our approach further before it can be empirically evaluated in a comprehensive fashion.

5.1. Metadata documentation

The first set of criteria is focused on the use of metadata as a basis for documentation. Table **5-1** summarizes our evaluation of metadata-oriented documentation.

Table 5-1: Metadata Documentation Criteria

Criteria	**Current Enterprise Data Architecture Approach**	**Tool Supported Metadata and Lineage**
Lineage	Operational Source to Data Warehouse only	Data Warehouse to Conceptual Business Model to Report
Meaning	in data warehouse and too far from the report	link report item element to data warehouse in a meaningful interactive fashion
Statistics	ad- hoc batch at best	live
Comments	Not available	Available online through the web application

- **Meaning and Lineage**

As mentioned in section 3.2.1, the meaning and lineage criteria were investigating if an approach is providing the correct and relevant descriptions and details about each and every data element in the data architecture so that a stakeholder who wishes to have a thorough grasp of a report could use the subordinate meaning and lineage documentation to gain the knowledge and insight they were looking for.

In enterprise data architecture, a manager who wishes to fully understand the context and details of a report can barely find someone who knows completely where each data element on each report is coming from. There are many different layers of data delivery in enterprise data architecture and different people are responsible for each part of architecture and they don't have time or knowledge to fully understand the other layers.

Therefore it might not be realistic to expect a data analyst to know or identify the lineage instantly. Even if possible, it is going to take quite a while to search all the layers of architecture to find the roots of the element. On the other hand, in the current practice of enterprise data

architecture, the lineage and data element tracking is only available at the operational source systems to the data warehouse level.

In contrast, according to the proposed framework, the right set of documentation is populated automatically by lineage mappings that cover the one to one mapping definitions throughout the whole spectrum of the data architecture i.e. from the report to report spec then to business conceptual models and from there to data warehouse and then to the operational source systems. The results of the metadata documentation efforts are all stored in an integrated and comprehensive database that is storing descriptive statements as well as binary files. The documents are available to the stakeholders on the pages of the proposed web application.

- **Statistics**

The next criterion is the availability of the statistical analysis, investigates the reliability of the statistical values based on the quality of providing up to the date results by counting the latest records in.

As mentioned earlier, statistics are a critical part of any reporting effort. Any report need to be analyzed quantitatively and statistical parameters and values provide the most common analysis. Although, the current practice produced some statistical analysis but they were ad-hoc and constrained to a limited span of time. These ad-hoc queries therefore, are not responsive to changes in data and as the records in a data warehouse change over time, they cannot adapt themselves automatically. In fact, the reporting effort should be repeated periodically.

Whereas in the proposed framework, we have built in formulas in the web- application which analyze whatever exists in the data warehouse in real time. As discussed in the previous

chapters, according to the automatic synchronization mechanism with the data warehouse, our metadata info is always up to date and these metadata provide the basis for the statistical built-in formulas in the application. Therefore, the statistical analysis is always in accordance with the latest changes in the data warehouse.

- **Comments**

The last criterion in this group reviews the means of communications and comment sharing in each approach. In the current practice, there was no structured and integrated system of comment sharing whereas in the proposed approach, the metadata documentation web application provides a mechanism to view and insert comments for each data item in the same place as the documentation of that element is presented.

5.2. Availability of documentation

Documentation, no matter how accurate or helpful, as long as is not easily accessible and reachable, wouldn't be employed by the end users since it does not add the desired value to the overall cycle. Table **5-2** summarizes our evaluation of the two approaches in this area.

Table 5-2: Availability of Documents Criteria

Criteria	**Current Enterprise Data Architecture Approach**	**Tool Supported Metadata and Lineage**
Who	2 to 4 experts + DBA	Anyone (assumes linked Metadata documentation web application understandable to usage)
Where	at the DW schema level	Report, conceptual, DW but not Data sources
When	static web site or update from Access DB i.e. often out of date or out of synch	From report , DW and web- application anytime
How	manually by 2-4 experts	point and click from report spec or manual copy search

- **“Who”**

The first measure in this category, which we named “Who”, seeks to find who has access to lineage information or can discover it. In the observed architecture in practice, only the database administrator who knows about all the entities of the data warehouse or few experts who have devoted their time specifically to lineage problem can realize the lineage. However, in our proposed architecture, anyone with some knowledge about the data and BI tools and applications can gain insight about the lineage and meaning of elements of a report. This claim is supported by the metadata documentation web application and report documentation mechanism that provide the documents and data item tracking regardless of time and location because of their automated nature.

- **“Where”**

Next we want to evaluate and contrast the availability of documents in regards to the different layers of the enterprise data architecture by the “where” criterion. We found out that in practice the data warehouse level of the data architecture is the only layer that has been explained and documented.

While, in the proposed architecture the documentation exceeds the data warehouse level only. We have documented every layer of the architecture i.e. reports layer, business conceptual models layer, and data warehouse. However, we have not placed more focus on the operational data source level, than what already exists in the enterprise data architecture.

- **“When”**

The “When” criterion compares approaches based on the time-limited accessibility to the up to date documentation as opposed to the case that the updated documentations are accessible at any time.

In current practice, there exists some online documentation which are mostly static HTML web-pages that at best are connected to a access database which needs to be manually updated every once in a while to get in synch with instantly changing data warehouse. On the other hand, the real time web application in the proposed approach provides unlimited (time wise) accessibility to up to date documentation from report layer of the architecture through data warehouse.

- **“How”**

Finally, “How” investigates about the mechanism that the documentations are accessed; are the documentations available through an automated process or an operator or clerk is required to present the documentations? In the enterprise data architecture this procedure is manual and is carried through by a group of experts.

In contrast, as we demonstrated in chapters 3 and 4, the proposed approach offers a point and click mechanism from report specification to extract the meaning and lineage through the automated metadata and report documentations applications. This point and click mechanism accounts for automated access to the documentation.

5.3. Documentation effort

This category looks into the effort needed to create, manage, and understand documentation. Table **5-3** summarizes our evaluation.

Table 5-3: Documentation Effort Criteria

Criteria	**Current Enterprise Data Architecture Approach**	**Tool Supported Metadata and Lineage**
Complexity, skill, difficulty	DBA+ domain expert	Anyone assuming that they can understand data
Duration	manual, weeks to run a report even for one report item, days	seconds to get documentation , minutes to a day to understand
Cost	DBA's and experts are expensive, opportunity cost, time valuable, reports not done, data misunderstood	investment in IT tool support, free after that
People	ad-hoc coordination of desperate people: user, domain expert, DBA	Still the same people but the system means users can self-serve.

- **Complexity, skill, difficulty**

As the first criterion in this group, we evaluate the complexity of the architecture, the diversity and depth of the skill and expertise to make use of the architecture, and the difficulties to do so. The current practice of enterprise data architecture relies on database administrator or a few domain experts to handle the problem. As explained in chapters 3 and 4, in the current practice of the enterprise data architecture, we need various experts in order to extract the meaning and lineage of the report elements.

While the proposed approach is designed so that anyone with the preliminary knowledge of data and methods to handle them can use the proposed toolset to bring meaning and lineage to the reports. Since the proposed approach contains automated specialized functions, they can

reduce the dependency to the human experts for each and every specific report. Also the documentation mechanism, maintains complete records of the previous procedures and efforts. Therefore, having thorough documentation of the previous efforts gives us the opportunity to reuse them in the following requirements

- **Duration**

The duration of the procedure is important as well. The time it takes from the moment any given element is observed on a report until its meaning and lineage info has been made available is very different between the two architectures under investigation. In the base case (the usual enterprise data architecture) it can take up to months to find the right documentation requested by stakeholders whereas in the proposed approach instant access to the right set of documentation is guaranteed by the point and click mechanism.

- **Cost**

Cost considerations are one of the key concerns of any organization. Basically, all the organizations seek to find ways to reduce costs and make a more productive use of the resources. Our research has been always towards satisfying these economizations. The proposed method is consists of a primitive investment in the IT tool support and a few hours of training. After that the service would be free.

Conversely, in the current practice every investigation into the meaning and lineage would cost hours of experts salary considering that the procedure would be long as well. Also since the documentations are not concentrated in one specific storage method and area, the need for new storage space increase over time. This alone, enforces more expenses to the system. The

lack of a good synchronization mechanism results in redundant efforts to prepare the same old reports only this time with the latest data. Therefore, the current practice cannot be cost efficient.

- **People**

The final question is about people interested in the lineage and meaning problem and how confident and smooth the procedure is in the architecture. The process of handling requests about the data and report elements semantics and lineage is meant to be ad-hoc in the current practice and requests are processed one by one and there would be temporary teams of domain experts assigned to each request.

In most cases the number of interested people would remain the same with or without our approach. However, as mentioned before, the proposed approach has a self-served mechanism for any authorized person who is interested to learn about the meanings and lineage roots of report and data elements.

5.4. Assumptions and Limitations

In this section we point out a few assumptions we made and limitations we encountered in practice throughout this research attempt. One of the key assumptions we made was the selection of the tools to create the conceptual business model and then the report specification for a report. We selected a set of theoretical tools which were the SDL, CDL, and MDL files and also we selected Cognos Report Studio to visualize out reports.

In previous chapters we talked about a series of mark-up language files called SDL, CDL, and MDL. As we mentioned earlier, they provide a theoretical and abstract model of the desired

reports in an XML format. But, since we are tackling a real industry problem in practice, we needed some visual and easy to understand tool. We used IBM Cognos products as the tool to generate managerial reports from our fake data. However, there are other options in the market as well. One of the other well-known applications is Microsoft Analysis Studio.

We pictured an approach in chapter three which aims at a fully automated and straight forward point and click mechanism to pull up the documentation about the semantics and lineage of the reporting elements. But then after we selected the back end reporting application which was Cognos Report studio, we encountered some conflictions and complexities to pair the resulting reports with our web application.

It is very complicated to access the report elements behind the user interface in such applications. Consequently, we decided to restrict our mechanism to one level lower and we actually started from a report spec down to the data warehouse and source systems documentations.

One of the biggest constraints to our research was data privacy policies and security. As a student obtaining access to specific items of the data warehouse and reports was either impossible or very difficult for some less critical pieces of information. Obviously, a healthcare organization is bound to treat the patient related data completely confidential.

Therefore, we could only access metadata and structural data without the actual records or even the statistics about the actual records. Even the stored procedures and views we developed to provide statistical values were never run in our presence.

On the other hand, conducting our research properly required records to for example generate some reports to see if our claims were coming true as we expected. As a matter of fact, we generated some records that were not real but close to reality using the data generation tools like Redgate SQL data generator[1].

Redgate SQL data generator is an application to quickly create realistic data based on a predefined SQL schema, provide demo databases without sharing live data, highlight potential improvements to databases, and future-proof databases by populating them with large quantities of test data. This way we were able to work on data that made sense and had nothing less than those stored in the hospital data warehouse.

[1] http://www.red-gate.com/products/sql-development/sql-data-compare/?utm_source=google&utm_medium=cpc&utm_content=brand_aware&utm_campaign=sqldatagenerator&gclid=CMmnkrTBqKgCFQnrKgod0D1OIQ

Chapter 6. Conclusion and Future work

6.1. Summary of Contributions

As mentioned earlier, the properties of a report documentation system are: to provide a clear view of the available reporting elements for further or more detailed reports; to facilitate the understanding of words and expressions in a report; to bring back the underlying data rows inside a database or data warehouse to the surface; and to provide a platform for the exchange of expert analytical ideas.

We proposed a framework that provides for all the aforementioned properties though conceptual business model of the report (or FM model), the data documentation tool, the mapping correspondents, and the documentation web application.

Contribution 1: A gap analysis of industrial practice and academic research in enterprise business analytics:

We started our research by conducting an organized series of interviews and meeting with data stewards working in the healthcare industry and pinpointing the opportunities to help them accomplish their task in an easier and faster manner as opposed to their existing procedures. Those meetings led us to the discovery of new user and stakeholder requirements and/or better presentation and interface design practices.

On the other hand, searching through the publications and previous works in the related area and lessons learnt from the past experiences taught us the best practices as well as gaps voids which has been overlooked or unexplained by the current practice and literature.

Finally, we articulated the identified gaps and ignored value creating features into a set of qualitative evaluation criteria in order to provide a base for improvements to the current industry practice and background literary works. At the end of the thesis, these evaluation criteria are employed to assess the proposed approach and toolset.

Contribution 2: A framework to provide and link systematic user-managed documentation for data management across the entire data architecture for business analytics:

In chapter 3 we offered an approach to bridge the gaps between the industry requirements and current practice. In our proposed approach we pictured our contributions to the enterprise data architecture by adding the proposed elements to the enterprise data architecture framework. Our approach incorporated a metadata repository that helps find the meaning, descriptions, and lineage of the data elements between different layers of the architecture. Also, the metadata repository helps the generation of conceptual business models by making all the data entities more visible and classified.

Furthermore, we pointed out a mechanism for documenting the lineage of the data elements through the various layers of the architecture. This mechanism provides for the tracing of the data elements within the whole data architecture spectrum. The documentation however was not fully automated but was stated in a mark-up language which provides a base for an automated report documentation system design and development.

Contribution 3: Tool Support for the framework that enables users to manage and use up to date documentation to understand data displayed in reports including

We developed a tool support including a web-based application to manipulate and update the metadata repository in real time. The proposed application is composed of methods, views, and stored procedures that update the metadata repository based on the latest changes in the data warehouse. Also, the web-pages provide for the point and click tracing mechanism between the conceptual business models all the way to the data warehouse and operational source systems.

Moreover, the proposed application calculated some statistical values for each and every table and column residing in the data warehouse in real time. The statistical functions however were revealed during the meetings with industry data stewards and the requirement gathering period.

Other features of the proposed web application include the facilities for comment sharing and applying the security considerations which were requested by the end users of the application. It's good to mention that the proposed toolset was fully developed and implemented in the healthcare organization under investigation.

The usability of any proposed tool is an important issue. Our proposed tool has not yet been formally evaluated along this dimension. However, as a proxy metric, let us consider user engagement. We are proud that our proposed tool has been put into production and is being used by the target users at the local hospital. These users have reported satisfaction with the way the tool implements their vision and preferences.

6.2. Future Work

6.2.1 Lineage Assessment

In our thesis attempt we tried to evaluate the overall proposed approach through a set of proposed qualitative evaluation criteria. As a future work, we suggest the generation of

quantitative criteria to evaluate the success of a lineage providing approach by quantifying the lineage maps and correspondents between the layers of the architecture.

Though, there have been studies which revolve around the evaluation of data lineage specifically. One of these studies introduces a methodology regarding the logical analysis of two proposed characteristics: completeness and well-behaved lineage (Benjelloun, et al. 2006).

6.2.2 Fully automated report documentation

In the chapter four we outlined the complex situation which led us to call off the development and implementation of a fully automated point and click tool to pull up lineage and meaning from the lower levels of the data architecture by a single click on any element on a report which is generated though a business analytics application.

As a future work, we suggest the design and development of a platform that can be integrated with any business analytics application to access the documentation and lineage maps in real time and in a reasonable amount of processing time.

References

[1] Bates , David W., et al. "Using information systems to measure and improve quality." *International Journal of Medical Informatics, Volume 53, Issues 2-3*, 1999.

[2] Azarm, Mana, Liam Peyton, and Fatemeh Nargesian. "Managing and Mapping Data Lineage for Business Intelligence and Analytics Applications in Health Care." *International Conference on Information Society.* London, UK: IEEE, 2011.

[3] Azvine, B, D Nauck, and C Ho. "Intelligent Business Analytics — A Tool to Build Decision-Support Systems for eBusinesses." *BT Technology Journal*, 2003.

[4] Batcheller, James K. . "Automating geospatial metadata generation—An integrated data management and documentation approach." *Computers & Geosciences Volume 34, Issue 4*, 2008.

[5] Bell , D, S Cesare , N Iacovelli , M Lycett, and A Merico. "A framework for deriving semantic web services." *Information Systems Frontiers, vol. 9, no. 1*, 2007.

[6] Benjelloun, Omar, Anish Das Sarma, Alon Halevy, and Jennifer Widom. "ULDBs: databases with uncertainty and lineage." *32nd international conference on Very large data bases.* Seoul, Korea: VLDB Endowment Inc., 2006.

[7] Berendt, Bettina, Sören Preibusch, and Maximilian Teltzrow. "A Privacy-Protecting Business-Analytics Service for On-Line Transactions." *International Journal of Electronic Commerce*, 2008.

[8] Bose, Ranjit . "Knowledge management-enabled health care management systems: capabilities, infrastructure, and decision-support." *Expert Systems with Applications, Volume 24, Issue 1*, 2003.

[9] Breault, Joseph L. , Colin R. Goodall, and Peter J. Fos. "Data mining a diabetic data warehouse." *Artificial Intelligence in Medicine 26*, 2002.

[10] Busborg, F, N Tryfona, and J. G. B. Christiansen. "A Conceptual Model for Data Warehouse Design." *ACM 2nd Int. Workshop on Data Warehousing and OLAP.* NY, USA: DOLAP, 1999.

[11] Carey, M. J., and D. J. DeWitt. "Of objects and databases: A decade of turmoil." *22nd VLDB Conference.* 1996. 3-15.

[12] Chaudhuri, Surajit , and Umeshwar Dayal. "An Overview of Data Warehousing and OLAP Technology." *ACM SIGMOD Record, Volume 26 Issue 1*, 1997.

[13] Cognos, IBM. *IBM Cognos 8 framework manager.* user guide, IBM, 2008.

[14] CUI , YINGWEI , and JENNIFER WIDOM. "Tracing the lineage of view data in a warehousing environment." *ACM Transactions on Database Systems (TODS), Volume 25 Issue 2*, 2000: 179 - 227.

[15] Cui , Yingwei , and Jennifer Widom. "Lineage tracing for general data warehouse transformations." *VLDB Journal*, 2003.

[16] Dayal, Umeshwar , Malu Castellanos, Alkis Simitsis, and Kevin Wilkinson. "Data Integration Flows for Business Intelligence." *12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '09).* NY, 2009.

[17] Dong, Jichang , Helen S. Du, Shouyang Wang, Kang Chen, and Xiaotie Deng. "A framework of Web-based Decision Support Systems for portfolio selection with OLAP and PVM." *Decision Support Systems, Volume 37, Issue 3*, 2004.

[18] Duan, A. X., C. Z. Chen, and T Li. *The Generation and Visualization of Cube EDM.* Vancouver, Canada: University of British Columbia, 2010.

[19] Dubois, Nancy, and Tricia Wilkerson. *Knowledge management: background paper for the development of a knowledge management strategy for public health in Canada.* Montreal, Que: National Collaborating Centre for Healthy Public Policy, 2008.

[20] Eder, Johann, and Christian Koncilia. "Changes of Dimension Data in Temporal Data Warehouses." *Lecture Notes in Computer Science, Volume 2114/2001*, 2001: 284-293.

[21] Fallman, Daniel. " Design-oriented Human–Computer Interaction." *Proceedings of Conference on Human Factors in Computing Systems, CHI 2003, CHI Letters, Vol. 5, Issue No. 1.* New York: NY: ACM Press, 2003. 225-232.

[22] Few, Stephen. *Information Dashboard Design: The effective Visual Communication of Data.* O'Reilly, 2006.

[23] Firestone, Joseph M. "Architectural evolution in datawarehousing and distributed knowledge management architecture." *White Paper, Executive Information Systems*, 1998.

[24] Gibson, Marcus , David Arnott, and Ilona Jagielska. "Evaluating the Intangible Benefits of Business Intelligence: Review & Research Agenda." *DSS2004.* Prato, Italy: Monash University, 2004.

[25] Greenberg, Jane. "Metadata Generation: Processes, People and Tools." *Bulletin of the American Society for Information Science and Technology, Volume 29, Issue 2*, 2003.

[26] Han, J, and M Kamber. *Data Mining: Concepts and Techniques.* USA: Morgan Kaufmann, 2006.

[27] Hevner, Alan R., Salvator T. March, Jinsoo Park, and Sudha Ram. "Design Science In Information Systems Research." *MIS Quarterly, vol. 28, no. 1*, 2004: 75-105.

[28] Holzinger, Andreas, Thomas Kleinberger, and Paul Müller. "Multimedia Learning Systems based on IEEE Learning Object Metadata (LOM)." *Proc. of ED-MEDIA 2001.* Tampere, Finland, 2001.

[29] Inmmon, William H . *Building the Data Warehouse.* Indianapolice: John Wiley & Sons, Inc., 2005.

[30] Inmon, William H. "The Data Warehouse and Data Mining." *COMMUNICATIONS OF THE ACM, Vol. 39, No. 11*, 1996.

[31] Kimball, R, and M Ross. *The Data Warehouse Toolkit.* New York: John Wiley & Sons, Inc., 2002.

[32] Kimball, R, M Ross, W Thornwaite, and J Mundy. *The Data Warehouse Lifecycle Toolkit.* New York: Wiley, 2008.

[33] KRULJ, DARKO , MILUTIN CUPIC, MILAN MARTIC, and MILIJA SUKNOVIC. "DATA WAREHOUSE MANAGEMENT SYSTEM-A CASE STUDY." *The 7th Balkan Conference on Operational Research, BACOR 05.* Constanta, Romania, 2005.

[34] Leitheiser, Robert L. "Data Quality in Health Care Data Warehouse Environments." *34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 6.* Maui, Hawaii: hicss, 2001.

[35] Liya, Wu, G Barash, and C Bartolini. "A Service-oriented Architecture for Business Intelligence." *SOCA '07 IEEE International Conference.* IEEE , 2007. 279-285.

[36] Lonnqvist, Antti, and Virpi Pirttimaki . "The Measurement of Business Intelligence ." *Information Systems Management Volume 23, Issue 1*, 2006: 32 - 40.

[37] Malinowski, E, and E Zimanyi. *Advanced Data Warehouse Design: From Coventional to Spatial and Temporal Applications.* Berlin: Springer, 2008.

[38] Meier, Marco, Werner Sinzig, and Peter Mertens. *Enterprise management with SAP SEM/business analytics.* Springer, 2005.

[39] Nargesian, Fatemeh. *Concept-Driven Multidimensional data modeling.* University of Ottawa, 2010.

[40] Pin, Peter, and Shan Chen. "The entity-relationship model—toward a unified view of data." *ACM Transactions on Database Systems (TODS)- Special issue: papers from the international conference on very large data bases*, 1976.

[41] Poole, John D. . "Model-Driven Architecture: Vision, Standards And Emerging Technologies." *Workshop on Metamodeling and Adaptive Object Models (ECOOP 2001).* Budapest, Hungary, 2001.

[42] Power, Daniel. "Decision Support Systems: From the Past to the Future." *AMCIS 2004* . NY, USA: AMCIS , 2004.

[43] Prather, J. C., D. F. Lobach, L. K. Goodwin, J. W. Hales, M. L. Hage, and W. E. Hammond. "Medical data mining: knowledge discovery in a clinical data warehouse." *AMIA Annu Fall Symp.* Washington, D.C.: PMC Journal, 1997. 101–105.

[44] Rao, Madanmohan . *Knowledge Management Tools and Techniques.* Elsevier Inc, 2005 .

[45] Rizzolo, Flavio, Iluju Kiringa, Rachel Pottinger, and Kwok Wong. *The Conceptual Integration Modeling Framework: Abstracting from the Multidimensional Model.* eprint arXiv:1009.0255, 2010.

[46] Rogers, Greg, and Ellen Joyner. "MINING YOUR DATA FOR HEALTHCARE QUALITY IMPROVEMENT." *SAS Conference Proceedings: SAS Users Group International.* San Diego, California: SAS Institute, Inc., 1997. 641-649.

[47] Scammell, Alison. *Handbook of Information Management.* London Taylor & Francis Routledge, 2001.

[48] Schubart, Jane R. , and Jonathan S. Einbinder. "Evaluation of a data warehouse in an academic health sciences center." *International Journal of Medical Informatics, Volume 60, Issue 3*, 2000: 319-333.

[49] Seufert, Andreas, and Josef Schiefer. "Enhanced Business Intelligence - Supporting Business Processes with Real-Time Business Analytics." *16th International Workshop on Database and Expert Systems Applications (DEXA'05).* Copenhagen, Denmark: DEXA, 2005. 919-925.

[50] Sim, I, et al. "Clinical decision support systems for the practice of evidence-based medicine." *Journal of the American Medical Informatics Association*, 2001: 527–534.

[51] Vaduva, A, and K. R Dittrich. "Metadata management for data warehousing: between vision and reality." *Database Engineering & Applications, International Symposium.* Grenoble , France, 2007. 129 - 135.

[52] Vercellis, Carlo . *Business Intelligence:Data Mining and Optimization for Decision Making.* Milano, Italy: John Wiley and Sons, Ltd., Publication, 2009.

[53] Vetterli, Thomas , Anca Vaduva, and Martin Staudt. "Metadata standards for data warehousing: open information model vs. common warehouse metadata." *ACM SIGMOD Record, Volume 29 Issue 3*, 2000.

[54] Williams, Steve, and Nancy Williams. "Assessing BI Readiness: A Key to BI ROI." *Business Intelligence Journal, Vol. 9*, 2004: 15-23.