Tool Support and Data Management for Business Analytics Applications in Healthcare

Mana Azarm, Fatemeh Nargesian, Liam Peyton
School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada

Abstract

The data delivery architectures in most enterprises are complex and under documented. When a healthcare 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. We propose a tool framework that includes: a metadata repository for enterprise data consolidated in the data warehouse; a systematic approach to data lineage mappings across the enterprise data architecture to link business analytics data back to the metadata repository; and tool support to manage metadata, link to it directly from reports, and keep metadata in synch with the data warehouse. We illustrate and evaluate our approach with a prototype implementation and a case study using a healthcare analytics dashboard for managing hospital acquired infections.

Keywords *- data lineage, metadata, business analytics, enterprise reporting*

1. Introduction

When a healthcare 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 values on a report make sense. Also, managers might not necessarily have the technical knowledge of software development needed to be able to generate their own customized reports.

The data delivery architectures in most enterprises are complex and under documented. Usually data is integrated from a variety of operational data sources into a consolidated data warehouse through ETL processes that extract, transform and load the data. To make this data accessible to end users, conceptual business models are created that provide simplified, and easy to navigate views of enterprise data for end users in the form of cubes or multi-dimensional data marts. Finally, business analytic applications and reports are built on top of these conceptual business models.

But the construction of such infrastructure is tedious, manually intensive to build, requires specialized technical expertise, and is especially difficult to map exactly where and how data on a screen came from in the organization.

In this paper, we investigate how a better understanding of data in business analytics applications can be addressed by tying metadata documentation and data lineage mappings to enterprise data architecture with tool support. We propose a tool framework that includes: a metadata repository for enterprise data consolidated in the data warehouse; a systematic approach to data lineage mappings across the enterprise data architecture to link business analytics data back to the data warehouse and its associated metadata repository; and tool support to provide appropriate interfaces to manage metadata, keep it in synch with the data warehouse, and link directly from reports data back to the appropriate metadata.

We illustrate and evaluate our approach with a prototype implementation and a case study using a healthcare analytics dashboard for managing hospital acquired infections that were developed in collaboration with data analysts at a large teaching hospital.

2. Background

In the following section, we are going to provide a brief overview of enterprise data architecture and related work on metadata documentation and data lineage mapping.

2.1. Data Warehouse and ETL

Data warehouses gather enterprise data from various operational data sources within an organization into a consolidated view to support business decision making [1]. 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 provide on-line analytical processing (OLAP) tools for the interactive analysis of multidimensional data of varied granularities [2]. Often reports and other presentation elements are packaged as business analytics applications.

2.2. Business Analytics Applications and Conceptual Business Models

Business analytic applications support users or knowledge workers in the role of data analysts and decision makers. Such systems can organize and present data in various formats in order to accommodate the diverse needs of different users. Usually, data is organized using a conceptual business model that presents a view of the data using business terms and objects familiar to end users.

Conceptual business models encapsulate the detailed features of underlying implemented databases [4]. Reports can provide snapshots of the performance of processes within an organization, in terms of the corresponding conceptual business model. A conceptual business model organizes the concepts into facts, hierarchies and dimensions in data cubes. In a business analytics application based on data cubes, reports are views or "slices of" data cubes. Reports consist of facts and measures aggregated along dimensions.

In our case study, we built a conceptual business model using the Conceptual Integration Model (CIM) framework [5]. An alternative to CIM is StarER [6] which creates a conceptual business model based on a set of user modeling requirements and concepts. Specifically, the StarER model integrates an ER (Entity Relationship) model and a star schema, widely used in data warehouses, with special types of relationships. The set of constructs in StarER accommodates: (1) facts, which represent measures of the underlying data warehouse, (2) entities, which show the objects of the environment which could correspond to levels in hierarchies [6], (3) attributes, specifying characteristics of entities, relationships or facts, and (4) relationships, which express the parent-child relations among entities and association of entities to facts.

Another alternative is Multidim [7] which classifies various kinds of hierarchies that exist in real world situations. It supports the representation of all multidimensional environment elements: dimensions, hierarchies, and measures. The graphical representation of this model is similar to an Entity-Relationship model [8]. Malinowski et al. proved that it is feasible to implement this model in current database management systems. Multidim [7] allows users to define simple, balanced, unbalanced, generalized, and a few complex hierarchies using the constructs in ER model together with a set of additional constructs.

Conceptual Integration Model (CIM) [5] offers a top-down framework in which users can conceptually abstract from an existing data warehouse. The conceptual model of CIM is called Conceptual Visual Language (CVL), which extends MultiDim [7] that in turn is based on Peter Chen's ER model [8] to support multidimensionality. The Store Visual Language (SVL) is the representation of the relational data warehouse schema, containing relational table definitions, keys and referential integrity constraints. Between the CVL and the SVL specifications, mappings are defined as a Mapping Visual Language (MVL) specification, which associate the conceptual business model to the columns of data warehouse tables.

2.3. Data Lineage and Report Documentation

Research on the connection between objects and databases has been underway for a long time [3]. In recent decades the problem has been tackled by bridging applications and databases using a set of lineage mappings. The lineage mappings associate data models used in applications to underlying databases.

By mapping the lineage of data, we can question the generated reports e.g.: What does the contents of a report mean? Where did the data come from? Are the data accurate? Do they mean what they should mean? Are they up to date? What are the characteristics of quality data? How to ask for data? How to generate reports?

"IBM InfoSphere Metadata Workbench [3] provides a web-based application for exploring data that is generated and used by InfoSphere Information server [4] applications and by external applications and processes.

The metadata workbench creates reports on data movement, data lineage, and the impact of changes and dependencies. In InfoSphere Metadata Workbench, it is possible to trace the data lineage of business intelligence reports to provide a basis for compliance with regulations such as Sarbanes-Oxley [5] and Basel II [6].

IBM InfoSphere Information Server components generate design-time, run-time, and glossary metadata and store it in the IBM InfoSphere Information Server metadata repository. Users can also import database and data file information into the metadata repository and create extended data sources and extension mappings that represent objects and processes that exist outside of IBM InfoSphere Information Server. With InfoSphere Metadata Workbench, users explore, analyze, and manage the metadata in IBM InfoSphere Information Server."

2.4. Web-based Documentation Tools

To have a thorough grasp of data in a warehouse which combines different sorts of data from 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" [1].

Metadata increases the speed and possibility of retrieving documents or pieces of data and helps the control and management of data [13]. Any documentation effort in this area should identify the table and column names, definitions, and also calculation rules and their attributes [14].

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 [2] through the documentation tool. The Metadata repository can be a SQL database with a relational schema. Metadata documentation is either generated by a human who is a professional metadata creator or generated automatically through machine processing of data. Metadata generation tools include intellectual tools, metadata standards, and technical compilations [15].

There are basically two categories of metadata management tools that can be considered: general-purpose versus model-based tools. General-purpose metadata management tools aim at enterprise wide data and try to document the whole system. These tools are meant to document and archive structures, systems and applications [16]. Model-based tools use metadata from a well understood schema of a specific data warehouse to achieve specific tasks. These tools are distributed between a data repository and software engine which come to participate together at runtime [16].

3. Tool Supported Metadata and Lineage

Our proposed approach is depicted in Figure 3. The basic enterprise data architecture (Operational Data Sources, ETL layer, Data Warehouse, Conceptual Business Model, Business Analytics Applications) is integrated with the following elements in order to provide metadata tool support for understanding data in business analytics applications:

1. Metadata Repository,
2. Lineage Mappings, and
3. Tool support

a. Metadata Documentation Web Application
b. Dynamic Synchronization with Data Warehouse
c. Report to Documentation Linkage.

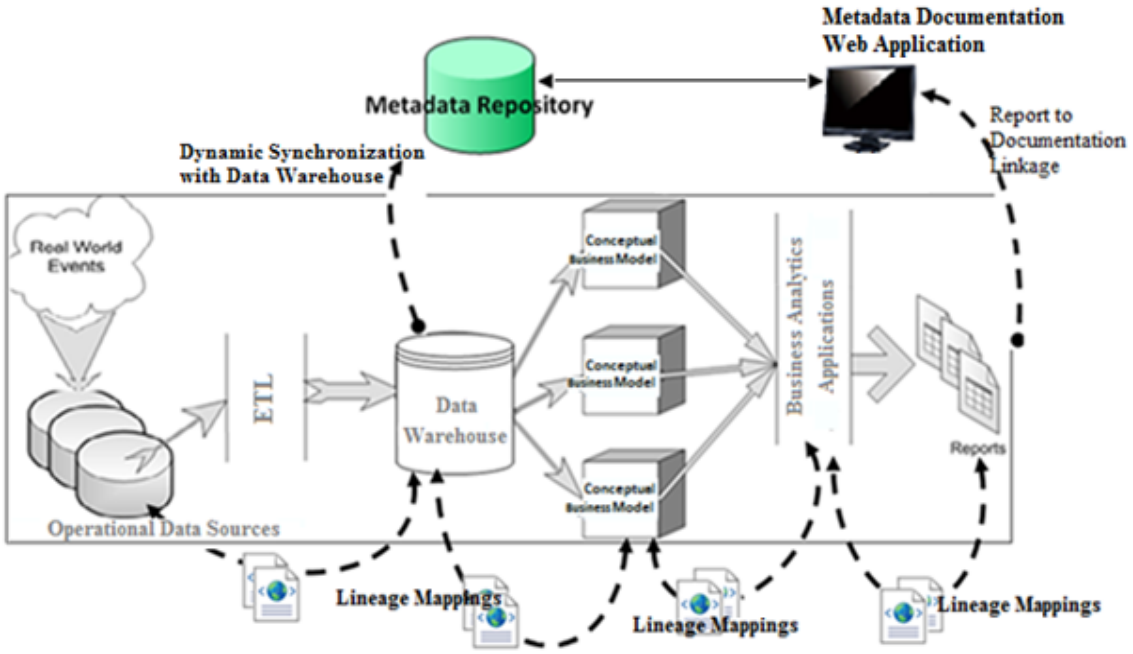

Figure 3. Elements of the proposed framework

3.1. 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 what is available and identifies each piece of data. Also like any type of data, metadata requires a persistent data store in which to keep it.

The metadata repository is designed to store the following data:

a) Comments: The comments posted on the proposed metadata documentation web application by end users are stored in the metadata repository,
b) Attachments: Another piece of information that some end users, like a data steward or stakeholder, might need to keep for future reference is a set of documents explaining the procedures and defining the related technical terms in the area. These documents in binary format are stored in the metadata repository,
c) Lineage: One on one entity mapping between the data warehouse entities and operational source systems entities is also stored in the metadata repository,
d) Statistics and usage information: Data analysts at the hospital use the usage and statistical analysis of values in data warehouse columns as a basis for reports and analysis. For example, the highest value in a column or the most frequent value, and
e) Typical metadata info: This includes information like short names, long names, data types, data formats, descriptions etc.

3.2. Data Lineage Mappings

As shown in Figure 2, our proposed mapping structure documents links between different layers of the enterprise data architecture in an automated fashion using a mapping correspondence. The initial mapping correspondence is based on a CIM framework which defines a conceptual model (using CVL) on top of the underlying data warehouse (modeled as SVL). The relationship between the conceptual model and data warehouse is captured by a set of mapping correspondences (defined in MVL). Looking at a report as a view on a cube with measures and hierarchies of descriptive data, each report can be documented by considering the portion of CVL which models the relevant measures and hierarchies. This way, the sources of data in the report can be traced by the mappings in MVL.

We then extend the approach to map between the conceptual business model and the report. The mappings at this level are based on mapping references to the business conceptual model contained within the report specification (which describes the specific queries that bind data to the report elments).

Figure 2 shows an example. In the "dataItemMeasure" section of the report spec, we see that Community MRSA is an entity of "Encounter" entity of the "Infection Control" data mart. The first mapping in Figure 3 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 (using MVL). This later mapping tells us that "Infection Control" in the conceptual business model maps to the "Nencounter" table in the data warehouse, and "CommunityMRSA", maps to a column in the data warehouse called "encCommMRSA".

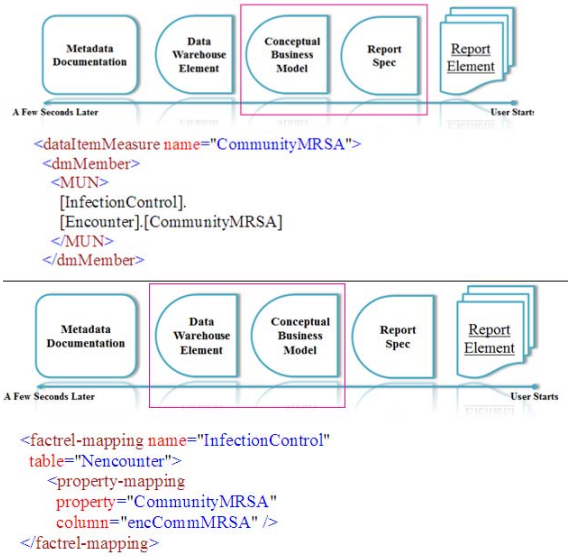

Figure 32. An example of report to documentation linkage

3.3. Tool Support

Tool support for our approach is composed of a metadata documentation web application, dynamic synchronization with the data warehouse, and support for report to documentation linkage.

The metadata documentation web application is designed to

a) present descriptive knowledge stored in the metadata repository in charted and well organized web pages,
b) provide means to download and upload documents and other binary files into the metadata repository,
c) share and store user comments,

d) establish a security and authentication mechanism to protect data privacy policies,
e) enable an administrator to modify user accounts, open new accounts, delete accounts and grant different read and/or write permissions,
f) update the metadata repository based on changes in the data warehouse.

To support synchronization of the metadata repository with changes in the data warehouse, we have created a mechanism integrated with our metadata documentation web application that compares the metadata tables in the metadata repository with data warehouse entities and updates the metadata repository to correspond to changes in the data warehouse.

Report to documentation linkage is provided by a tool to pull up relevant metadata documentation for any report element viewed by a user with no technical expertise required. The proposed tool automates the flow of control (illustrated in Figure 3) 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).

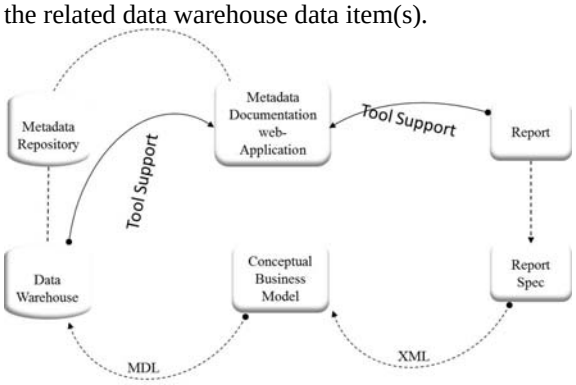

Figure 33. Lineage offered by the tool support

4. Case Description: Healthcare Data

Healthcare presents various interesting data warehouse design opportunities and situations. The data related to a typical large healthcare organization are shared among diverse organizations [9]. Health related data arrives from various organizations such as clinics, laboratories, physician's offices, hospitals, etc. We first give an overview of current approaches and problems with metadata documentation before showing how our approach addresses the problem.

4.1. Current approach to metadata documentation and data lineage at the local hospital

Data coming from all the aforementioned sources is extracted, transformed, and sorted into an integrated data warehouse to provide a platform for quality of healthcare, data analysis, and performance assessment. To extract reports from this giant data warehouse, we need to understand exactly what is stored in it. Metadata documentation can help organize and understand data within such a repository. Information processing from source to reports in such an organization is summarized in Figure 4.

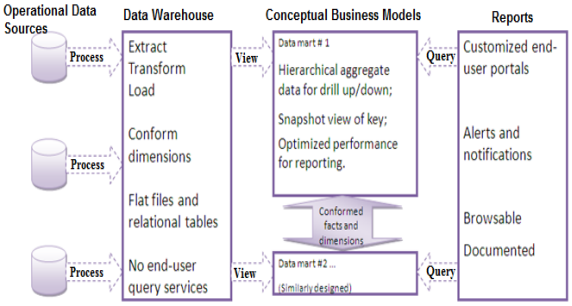

Figure 44. Relationship between data warehouse elements

At the local hospital under investigation, like any other healthcare organization, there is a huge amount of data stored in the data warehouse. The reporting procedure starts when it's time for a new version of a periodic report as a performance measurement tool or when a user specifically asks for a new report. Quite often a user may ask for further documentation to understand the components and implications of a report.

Data analysts are responsible for generating reports and managing the supporting documentation. They often have to collaborate with multiple domain experts and DBAs to help them understand the data items required for a report, their description and source information. The reporting process 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.

The data analysts we worked with at the hospital documented their efforts and accomplishments 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 missing as well, often leading to critical gaps in knowledge needed to understand how a value was obtained on a report.

4.2. Infection Control Report

We now illustrate the advantages of our approach with an example. We start from the report that a user can see on a screen and show how using our data lineage framework, the user gains insight about the semantics of the report elements, their roots to the data warehouse and knowledge of the elements available for further reporting needs. In our example, we used IBM Cognos Framework Manager (FM) to build the conceptual model in the context of a CIM framework and we used IBM Cognos Report Studio, to generate the infection control report based on the FM model.

Fig. 5 shows a screen shot of the infection control report in which 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 documentation about the elements of the report like "CommunityMRSA" (circled).

In our example healthcare case, we generate "Hospital and Community acquired MRSA Rates" as a periodic report. "Community acquired MRSA rate" is one of the key data elements on that report which is calculated through taking the sum of community acquired MRSA's.

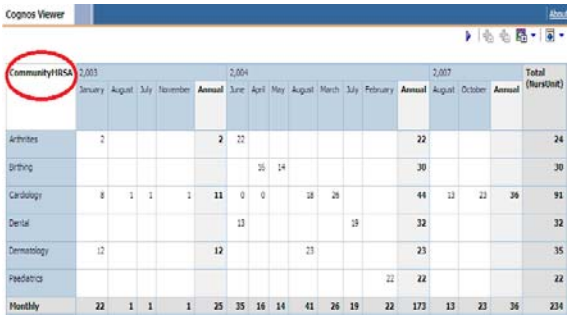

Figure 45. Screen shot of example report in IBM Cognos Report Viewer

IBM Cognos Report Studio defines the layout and queries for a report in an XML formatted Report Specification. In the Report Specification, community-acquired MRSA rate can be spotted in the "data item measure" tag. Below, we can see an excerpt from the Report Specification for our example report. When this XML file is parsed into our documentation tool, its relation to the data warehouse is captured through our extension to the CIM framework and the user can find more detailed info about the column feeding the report on the warehouse documentation as shown in Fig. 6.

Report Specification:

```
<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 Report Specification, CommunityMRSA is a data item measure that comes from the Encounter entity and dimension on the conceptual model which is connected to the Infection Control fact on its behalf. This relationship can be found on the conceptual model map in Figure 6. However, what this means and what it is made up of is the question we are going to answer through our lineage structure.

Typically a report contains values that measure some critical performance indicators in order to facilitate decision making. Each performance indicator is calculated through a formula whose parameters are fed from the data elements within a data warehouse.

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". Using the standard dimensions in healthcare [9] and performing customized analysis on data requirements of the hospital, we selected the following dimensions: Encounter, services, patient, encounter start date, encounter end date, facility and providers.

As we mentioned earlier we use dimensional data-marts to help generate reports. The Conceptual Visual Model (CVL) shown in Figure 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.

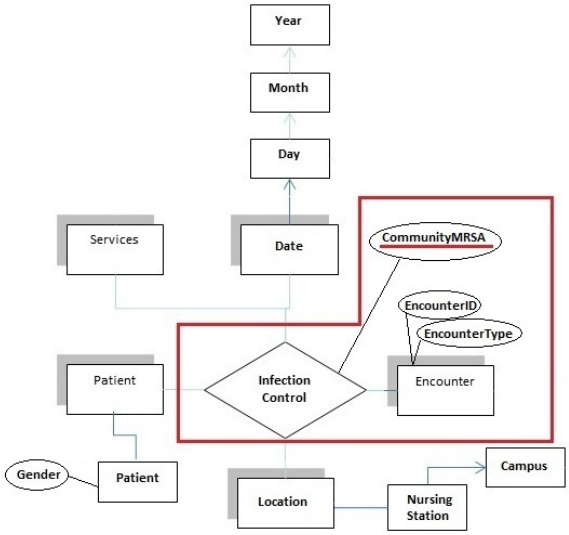

Figure 6. Our conceptual model for the healthcare example

4.3. Data Lineage using MVL

As mentioned in section 3, MVL associates the conceptual business model (CVL model) to the data warehouse (SVL model). Simple attribute-to-attribute correspondences or mappings are defined between the columns in data warehouse tables and attributes in the conceptual model. These mappings allow great flexibility in the data that can be mapped together. The mapping decisions are made based on what information is required to be seen in data lineage reports.

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

The CIM framework provides a mapping compiler which takes as input a pair of CVL and SVL schemas and an MVL mapping between them. Based on the mapping fragments and the integrity constraints in the schemas, the compilation process generates a set of views that map conceptual model elements to database tables and columns. This ultimately allows users to have an integrated view of underlying data, especially for the cases where data is scattered in different tables. In the generated view queries, tables are joined through their foreign keys. This provides users with additional knowledge of semantic relationships among tables. Also, it declares the possibly complicated relation of tables involved, as a whole, to a conceptual element. In the case of fact relationship views (measures), the view can specify the way a number on the report is calculated from data in tables.

In our Infection Control report, the elements such as fact table, dimensions and hierarchies are documented within MVL files which explain one on one mappings between report elements and data warehouse columns. Note that such mappings have a generic context that can be applied to any reporting endeavor. As a manager or stakeholder reviews these documents, they would understand each building block of the report.

In the schema of our example data warehouse, “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 model. The mappings, defined in MVL, provide clues on how the data in data marts and reports are related to the tables of underlying data warehouses. The Infection Control MVL is presented below.

MVL:

```
<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>
```

As we can see in the MVL file, CommunityMRSA is the report property that maps to the “encCommMRSA” column within the “Nencounter” table in the DB schema and also the report called “InfectionControl”. Using our web-based tool, we are able to search the MVL files for specific report elements and then read the documentation about the column(s).

4.4. Infection Control Metadata Documentation

The metadata documentation was initially proposed to address these issues:

a) to provide details about the column and what it’s intended to capture,
b) to provide info about the source system injecting the records,
c) to calculate some statistical variables over the values of each column, and

d) to provide a base for communication by users about data and a means to save comments about each data element.

Also, according to our proposed framework, the metadata repository helps understand the data within the data warehouse and therefore helps to recognize the right fact table and dimensions.

Our metadata repository contains technical metadata as it defines the objects that make up the data warehouse from a technical perspective. It includes system metadata defining the data structure e.g. tables, fields, data types, indexes, etc. It also describes the contents of the data warehouse in more user accessible terms. It tells us what data we have, where they come from, what they mean, and what relationship they have to other pieces of data in the warehouse.

We also provide the user with some classifications and grouping of metadata objects and their analysis e.g. minimum value, median value, top 10 values, etc. related to each column. The descriptions and name tags for each column in the data warehouse are defined and gathered in the specified metadata columns. The metadata repository contains some tables that provide statistical analysis over the values stored in each attribute of a data warehouse as well as clarifying the source system the data element is coming from. The statistical analysis helps managers realize their critical and sensitive areas and the need for capturing periodic reports about them.

Table 1. Example metadata documentation info provided by metadata documentation web application

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	
Missing values	60
Variable Class	Discrete
Minimum value	0
Maximum value	1
Top 1 value	1
Top 1 frequency	1

The web application also provides a means to write comments or notes about each column. This is specifically useful by data analysts and system developers to share their experience and/or knowledge. Data analysts are also able to share files and usability guides. Table 1 shows some of the information provided by the web application.

5. Evaluation

Table 52 summarizes the benefits of using the proposed framework compared to current enterprise data architecture.

Table 52. Proposed approach vs. enterprise data architecture

Measure	Enterprise Data Architecture	Proposed Approach
Lineage	source in data warehouse only	Data Warehouse to Conceptual Model to Report
Statistics	ad- hoc batch at best	live
Meaning	in data warehouse and too far from the report	link report item element to data warehouse in a meaningful interactive fashion
Comments	Not available	Available online through the web-application
Who	2 to 4 experts + DBA	Anyone (assumes linked DW documentation understandable to usage)
Where	at the DW schema level	Report, conceptual, DW but not Operational 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
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	Same people but the users can self-serve.

In current 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 the time or knowledge to fully understand the other layers. Therefore it is not 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.

However, in the proposed framework the lineage process takes a few seconds and can be done by any authorized person. Also the web-based application provides means for sharing and adding comments for each specific column which acts as a base for communication.

6. Conclusion and Future Work

In our research, we try to facilitate the clear lineage from a report to its corresponding conceptual business model and the source data elements in the data warehouse. As mentioned earlier, the properties of a metadata 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 using a metadata repository, data lineage mappings and tool support in the form of a web application for managing documentation, synchronization with the data warehouse and direct linkage from report elements to the corresponding metadata in the data warehouse.

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 characteristics: completeness and well-behaved lineage [10]. However, the evaluation of different data management techniques and framework is an area for research and study in future work.

7. Acknowledgment

This work was supported by a CHRP grant from NSERC and CIHR for "Performance Management at the Point of Care".

8. References

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

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

[3] N. Tryfona, F. Busborg and J. G. B. Christian, "starER: A Conceptual Model for Data Warehouse Design," in *ACM 2nd Int. Workshop on Data Warehousing and OLAP (DOLAP)*, Kansas City, Missouri, USA, 1999.

[4] F. Rizzolo, I. Kiringa, R. Pottinger and K. Wo, "The Conceptual Integration Modeling Framework: Abstracting from the Multidimensional Model," Cornell University Library, 1 Sep 2010. [Online]. Available: http://arxiv.org/abs/1009.0255. [Accessed 12 December 2010].

[5] D. Linstedt, "Data Vault Modeling and Methodology," [Online]. Available: http://danlinstedt.com/about/data-vault-basics/. [Accessed 10 January 2011].

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

[7] P. Chen, "The entity-relationship model toward a unified view of data," *ACM Transactions on database systems*, vol. 1, no. 1, pp. 9-36, 1976.

[8] M. Carey and D. DeWitt, "Of objects and databases: A decade of turmoil," in *Proceedings of the 22nd VLDB Conference*, Mumbai, India, 1996.

[9] "IBM InfoSphere Metadata Workbench," 2008. [Online]. Available: http://www-01.ibm.com/software/data/infosphere/metadata-workbench/. [Accessed 02 02 2011].

[10] "IBM InfoSphere Information Server," [Online]. Available: http://www-01.ibm.com/software/data/integration/info_server/. [Accessed 02 02 2011].

[11] "The Sarbanes-Oxley Act," 2002. [Online]. Available: http://www.soxlaw.com/. [Accessed 04 02 2011].

[12] "Basel II: Revised international capital framework," 2004. [Online]. Available: http://www.bis.org/publ/bcbsca.htm. [Accessed 04 02 2011].

[13] R. Kimball, M. Ross, W. Thornwaite and J. Mundy, The Data Warehouse Lifecycle Toolkit, USA: Wiley, 2008.

[14] A. Holzinger, T. Kleinberger and P. Müller, "Multimedia Learning Systems based on IEEE Learning Object Metadata (LOM)," in *Proc. of*

ED-MEDIA 2001, Tampere, Finland, 2001.

[15] J. L. Breault, C. R. Goodall and P. J. Fos, "Data mining a diabetic data warehouse," *Artificial Intelligence in Medicine 26*, pp. 37-54, 2002.

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

[17] A. K. R. D. Vaduva, "Metadata management for data warehousing: between vision and reality," in *Database Engineering & Applications, International Symposium*, Grenoble , France, 2007.

[18] R. Kimball and M. Ross, The Data Warehouse Toolkit, USA: Wiley, 2002.

[19] O. Benjelloun, A. D. Sarma , A. Halevy and J. Widom, "ULDBs: Databases with Uncertainty and Lineage," in *VLDB 06*, Seoul, Korea, 2006.

