A DECISION SUPPORT PROTOTYPE FOR THE KANSAS STATE UNIVERSITY LIBRARIES

Maria Zamfir Bleyberg, Raghunath Mysore, Dongsheng Zhu, Radhika Bodapatla
Computing and Information Sciences Department

Karen Cole, Michael Somers
Hale Library

Kansas State University, Manhattan, KS, 66506

Abstract

In this work, we present a prototype of a decision support system, which has been built at Kansas State University, to assist the university library administration make informed decisions regarding the acquisition of books and the subscriptions and cancellations of serials.

Keywords: library data warehousing, data mining, decision support systems

1 Introduction

A data warehouse is a repository of integrated information from distributed, autonomous, and possibly heterogeneous, sources [3, 7]. The principal purpose of a data warehouse is to provide information to the management of an enterprise for strategic decision making.

Users use front-end tools to interact with the data warehouse. Managed query, on line analytical processing (OLAP), and data mining are basic decision tools. Managed query tools shield end users from the complexities of SQL and database structures by providing subject-oriented views of a database and supporting point-and-click creation of SQL queries. OLAP tools provide an intuitive way to view corporate data. These tools aggregate data along common business subjects or dimensions and then let users navigate through the hierarchies and dimensions with the click of a mouse button. Users can drill down, across, or up level in each dimension. Data mining tools provide insights into corporate data that are not easily discerned with managed query or OLAP tools. They are used to extract implicit, previously unknown and potentially useful patterns from data in the data warehouse [1, 2].

Because of a limited budget, the administration of the Kansas State University Libraries (KSUL) must select carefully what books to acquire and what serials to subscribe to. The KSUL administration can benefit from a decision support system that will help them operate more effectively their funds and satisfy all the requests for books, journals and other periodicals of faculty, students, and other library users (patrons, for short).

Let us consider the following scenario: a patron goes to the library looking for a specific book or a journal article. The following possibilities exist:

- The library has the requested material; if it is a journal, it means that the library has a subscription to the journal, or, that the library has a contract with an on-line vendor who can provide the journal article.
- The library does not have the requested material. In this case, the material could be obtained by either purchasing it or by borrowing it for a limited time from other libraries.

If the same book or journal is borrowed several times, the cost of borrowing it from other libraries may exceed the cost of purchasing it. The analysis of the usage patterns of the books and journals over a certain period of time may help the KSUL administration decide when to purchase a book or get a subscription to a journal, and not to borrow it. The patterns could also show the departments and people who use heavily the library services. Such an analysis can be successfully accomplished only by building a decision support system for KSUL.

2 The Architecture of the Decision Support Prototype

A data warehouse is the core of any decision support system. In Figure 1, we outline the basic architecture of a warehouse: data is collected from each source, in-

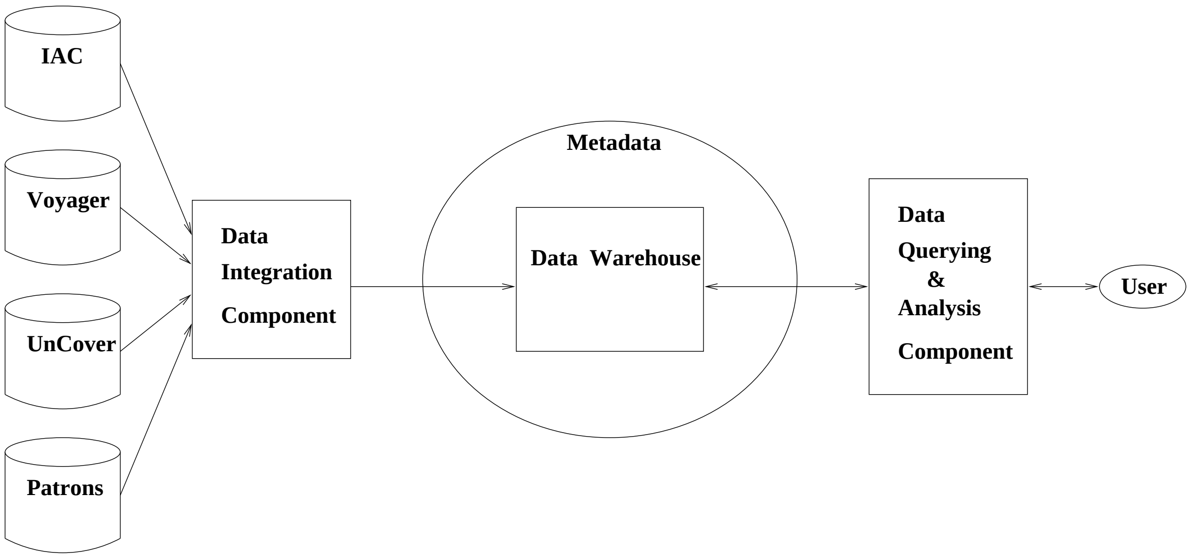

Figure 1: The architecture of the data warehouse prototype

tegrated with data from other sources, and stored at the warehouse.

In this work, we adopted a declarative approach to integration [4]. This approach is based on building a conceptual representation of both the information sources and the data warehouse. An important aspect of the conceptual representation is the explicit specification of the set of interdependencies between objects in the sources and the objects in the data warehouse.

Information integration can be either *virtual* or *materialized*. In the first case, the integration system acts as an interface between the user and the sources. In the second case, the integrated information in a persistent store is called a data warehouse.

The methodology for source integration in the data warehouse deals with two scenarios: source-driven integration and client-driven integration. Source-driven integration is triggered when a new source is taken in consideration for integration. The client-driven design strategy refers to the case when a new query posed by a client is considered.

The current decision support prototype for KSUL has data collected from the following sources:

- **Voyager**,
- **Human_Resources** (**HR**),
- **Student_Information_System** (**SIS**),
- **ILL**, inter-library loan,
- **IAC** (an abstracts, citations, and full texts on-line database)
- **UnCover** (an on-line document delivery vendor)

Voyager is the KSUL operational database, which maintains records of the daily transactions of books, journals, and all the information pertaining to acquiring and lending of the library documents.

The KSUL has contracts with vendor companies, such as **IAC** and **UnCover**, which offer on-line services, consisting of searching and retrieving of articles from various journals. These vendor companies keep records of the usage of their services by patrons and the cost of the services rendered.

An **IAC** record contains monthly summary of the usage of a specific journal. It does not contain the patron name who uses **IAC**. A patron uses **IAC** by entering a citation of an article; such a usage is counted as a *view*. If a patron chooses to print or download the article, then such a usage is classified as a *transaction*. If only the abstract of the article is retrieved, then such a usage is classified as *abstracts*. There is an annual subscription cost for **IAC**.

UnCover provides a searchable citation database. **UnCover** delivers the requested articles (from journals) to patrons either by FAX or regular mail. **UnCover** has transaction as the only type of usage. A typical monthly report from **UnCover** contains information about the journal title, article title, author(s), patron name, and cost (which includes copyright fee plus delivery fee). **UnCover** provides articles from 1500 journals.

The **HR** and **SIS** databases contain all the information pertaining to faculty, students, and other employees of the KSU.

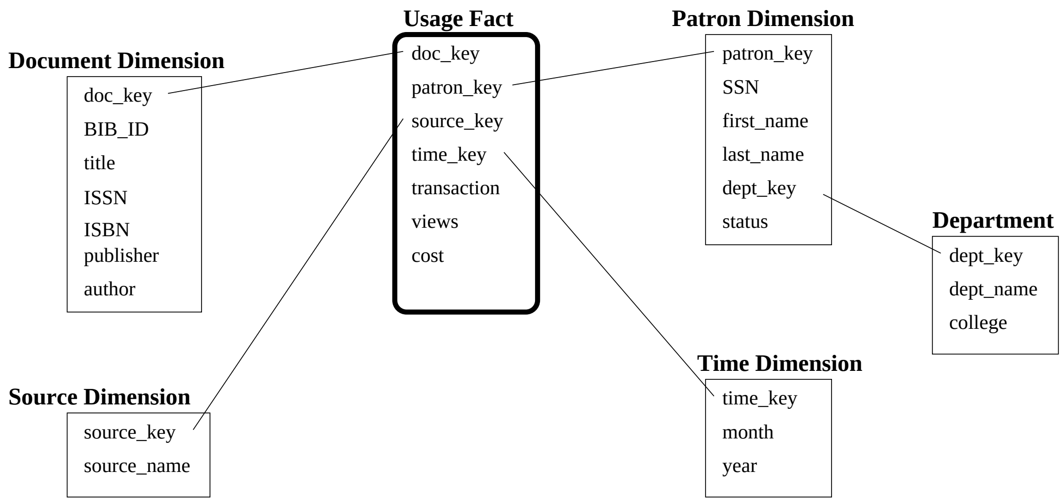

Figure 2: Star join schema of the data warehouse

The **ILL** database contains all the information regarding the material borrowed from other libraries, which includes the name of the patron who requested the material, the name of the library that lent the material, the title of the material, the borrowing and return dates, and the cost (shipping and handling).

The records of the data sources described above have hundreds of attributes. We used the client-driven design strategy, the discussions with the KSUL administration, and the analysis of the queries posed by them, to select the attributes for the decision support prototype. We used the relational model to represent both the data sources and the data warehouse database. The attributes of the underlying operational database sources that have a role in the current prototype are shown below.

Voyager(*patron_first_name, patron_last_name, SSN, journal_title, author, ISSN, ISBN, BIB_ID, transaction, borrowing_date, publisher, cost*)

HR(*employee_first_name, employee_last_name, SSN, college, department, status*)

SIS(*student_first_name, student_last_name, SSN, college, major_dept, status*)

ILL(*patron_first_name, patron_last_name, journal_title, ISSN, library_name, transaction, borrowing_date, cost*)

IAC(*journal_title, ISSN, views, transaction, abstracts*)

UnCover(*journal_title, article-title, author, patron_first_name, patron_last_name, transaction, cost*)

Attribute abbreviations are explained below.

SSN	- Social Security Number,
ISSN	- International Standard Serial Number,
ISBN	- International Standard Bibliographical Number,
BIB_ID	- Bibliographical Identification Number

In the current prototype, we did not include data on books, in order to keep its size under control.

The conceptual representation of the data warehouse is given in the next section.

3 The Multidimensional Model of the Data Warehouse

In the broadest sense, a data warehouse refers to a single, integrated database that contains very large stores of historical data. To ensure easy access to this vast amount of data, modern data warehouses typically adopt a dimensional approach to information processing instead of a traditional relational database approach. Unlike the entity-relational model, the dimensional model is very asymmetric. In this model, data is divided into two categories: *facts* and *dimensions*. Facts are the core data elements being analysed, and dimensions are attributes about facts. This method of representing data is known as star schema. The facts are represented as a table in the center of the schema. It is the only table in the schema with multiple joins connecting it to the dimension tables. Facts are almost

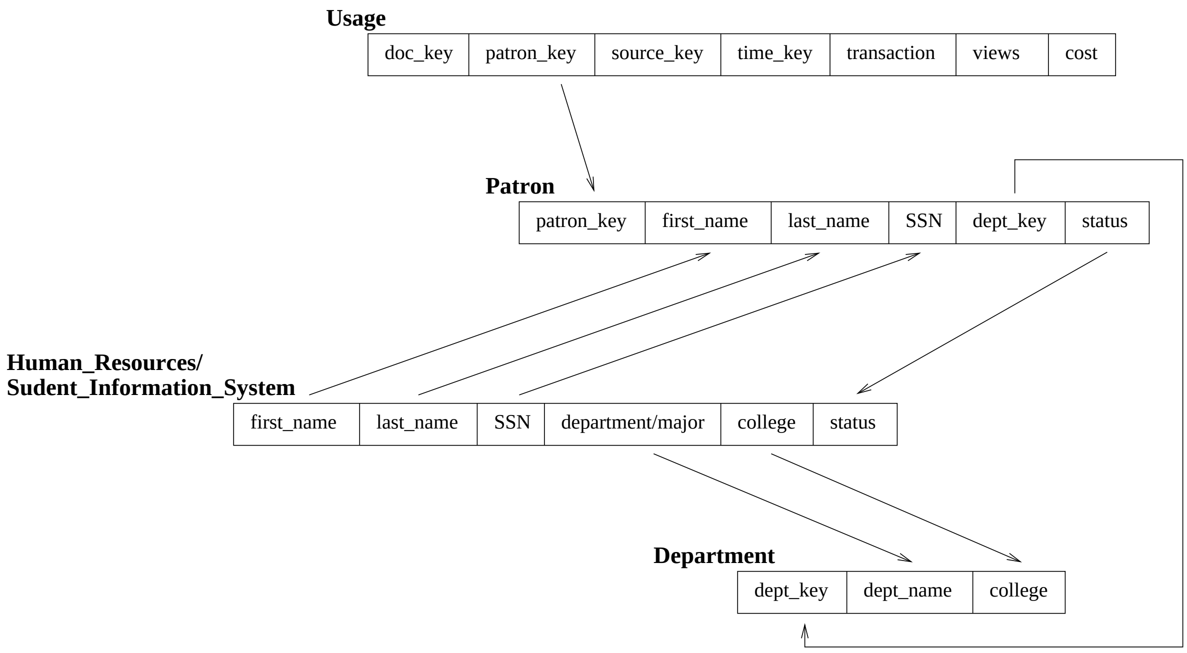

Figure 3: Interdependencies among data source and data warehouse attributes

always numeric and additive. The fact table is heavily populated compared to the surrounding dimension tables.

In the current data warehouse prototype, the *cost* and *usage* of the journals are the core elements of the fact table. The usage of a journal is represented by two attributes, *transaction* and *views*. The *transaction* attribute keeps track of the journals that are checked out from the library and of the on_line articles that have been downloaded. The on_line reading of a journal article is recorded as a *view*. The grain of the fact table is the *monthly* usage of the KSUL periodicals, because the records of the database sources **IAC** and **UnCover** contain only the monthly usage of the periodicals.

Figure 2 shows the star join schema of the warehouse database, which has one fact table and four dimensions. Some dimensions have hierarchical relationships, such as:

- **Document**: publisher, title.
- **Patron**: college, department.
- **Time**: year, month.

We chose the Oracle database management system on a UNIX platform to implement the warehouse database. In order to use this implementation environment, we mapped the star join schema into a relational database schema. A relation has been created in Oracle for each dimension and fact table of the star join schema, preserving all the integrity, referential and semantic constraints imposed by the multidimensional model. The loading of data from the source databases into the warehouse database is discussed in the next section.

4 Data Integration

According to [6], integration is the most important aspect of a data warehouse. When data passes from the application-oriented operational environment to the data warehouse, possible inconsistencies and redundancies should be solved, so that the warehouse is able to provide an integrated and reconciled view of data.

The cleaning operation detects noisy and incomplete source data and provides solutions for source integration. We present in Figure 3 part of the interdependencies among the data source and data warehouse attributes. In our source data, we found many violations of these interdependencies:

- absence of attribute values,
- domain inconsistencies,
- duplication of records,
- non-unique identifiers.

For example, we detected that the prototype attributes *cost*, *status*, and *college* get values from data source records that may have missing values. Because these attributes are essential for the formulation of decision queries, we inserted individually the missing values.

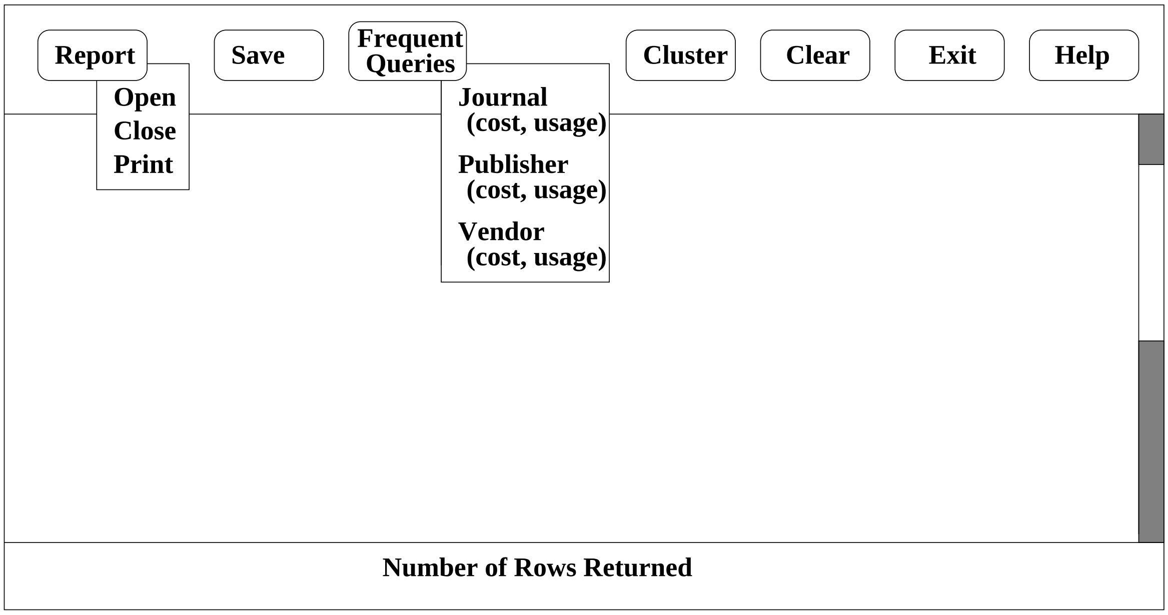

Figure 4: Graphical user interface

A careful examination of the departments at KSU led to 209 distinct department names. That was in contradiction with 440 distinct department names found in the **HR** and **SIS** operational databases. We solved this problem by clustering the 437 names and then classifying the clusters according to the 209 names. We developed a tool that detects and updates automatically the incorrect department names.

The analysis of the patron data from 7564 records of the **HR** and **SIS** databases revealed 150 people with duplicate records. These people are graduate students who also work in different departments. In the prototype, we used only one record for each of these people.

5 Decision Support Tools

The prototype provides a graphical user interface (GUI) (see Figure 4) to the user, which displays the available decision support tools. The Java object-oriented programming language with embedded SQL has been used for this purpose. Queries are defined in a subset of SQL that includes select-project-join and aggregate operations over all of the warehouse relations.

The data analysis tools considered in the current prototype include managed query, OLAP, and clustering techniques. The following are examples of frequent queries:

1. Find the cost (and/or usage) of a given journal for a given period (one month, a sequence of months in a year, one year).

   Example : Obtain the costs and usage of a serial whose title is 'The New York Times' for the period July, August, September, 1996.

2. Find the cost (and/or usage) of the serial(s) published by a specific publisher, for a given period (one month, a sequence of months in a year, one year).

3. Find the cost (and/or usage) of the serial(s) published by a specific publisher and supplied by a particular vendor, for a given period (one month, a sequence of months in a year, one year).

4. Find the list of all the journal with the minimum (or maximum) number of transactions for a given period (one month, a sequence of months in a year, one year).

5. Cluster the data warehouse records using the attribute *journal_title* (or any other attribute).

Clustering is the task of segmenting a heterogeneous population into a number of more homogeneous subgroups or *clusters*. In clustering, there are no predefined classes. The records are grouped together on the basis of self-similarity.

In the current prototype, we used the pattern-based knowledge induction (PKI) technique [8] to cluster attribute values in the database. *Patterns* are conditions on attributes values, such as:

patron_last_name = "Smith", or
journal_title = "Computer"

A *rule* is an inferential relationship between two patterns A and B, represented by A $\rightarrow$ B, indicating that when A is true, B also holds. For example,

$$patron\_name = \text{“Smith”} \rightarrow journal\_title = \text{“Computer”}$$

The PKI algorithm groups attribute values which appear as premises in rules with the same consequence. For example, if the attribute *patron_last_name* is selected, all patrons who requested articles from the same journal for equal cost are clustered together.

6 Summary and Future Work

In this paper, we presented the design and implementation of a decision support prototype for the libraries of the Kansas State University. The primary goal of this system is to help the library administration decide when to purchase subscriptions to certain periodicals.

A data warehouse system cannot be built in one cycle. The typical approach is to use multiple small development cycles [5]. Work in progress includes the extension of the current prototype by adding new data sources to the data warehouse. The loading of the new data into the warehouse poses interesting problems, such as, the development of criteria that indicates when the existing target data must be *replaced* by the new data, when the existing target data and new data must *merge*, and when the new data must be *appended* to the existing data. The addition of new data to the warehouse database will make possible to consider new queries and therefore, new decision tools must be implemented.

References

[1] P. Adriaans and D. Zantinge. *Data Mining.* Addison-Wesley, 1997.

[2] M.J.A. Berry and G. Linoff. *Data Mining Techniques.* John Wiley and Sons, Inc, 1997.

[3] A. Berson and S. Smith. *Data Warehousing, Data Mining, and OLAP.* McGraw-Hill, 1997.

[4] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Source integration in data warehousing. In *Proc. 6th Int. Conf. on Cooperative Information Systems*, 1998.

[5] B. Devlin. *Data Warehouse from Architecture to Implementation.* Addison-Wesley Longman, 1997.

[6] W. H. Inmon. *Building the Data Warehouse.* John Wiley and Sons, Inc, 1996.

[7] Ralph Kimball. *The Data Warehouse Toolkit.* John Wiley and Sons, Inc, 1997.

[8] M. Merzbacher and W. Chu. Pattern-based clustering for database attribute values. In *Proc. AAAI Workshop on Knowledge Discovery in Databases*, 1993.