REAL TIME SCIENTIFIC DATA ARCHITECTURE FOR INDIAN SCIENTIFIC COMMUNITY

JAYANTHI RANJAN
Professor in Systems and IT, Institute of Management Technology, Raj Nagar, Ghaziabad.
Uttar Pradesh, INDIA

E-mail: jranjan@imt.edu

ABSTRACT

One of the most complex and time-consuming problems facing in the design of a data warehouse is to extract, transform, and load different operational data sources into an integrated data warehouse. Building a data management model for scientific applications poses challenges that are not normally encountered in commercial database development. Complex relationships and data types, evolving schema, and large data volumes (in terabyte) are some commonly cited challenges with scientific data. In this paper, we propose a data warehouse framework architecture model thru XML to manage and analyze scientific behavioral data. Streams of data pour in from the scientific data particularly from satellites especially from data related to weather and other geo-spatial details amounts to terabytes of data each year from each satellite. These data if properly managed can be of immense value both to commercial community as well as scientific community. Here we have proposed an architectural schema through which scientific data can be archived. This architecture will try to have a process defined thru which easy retrieval of metadata is possible.

Key Words: *Scientific Data, Schema, Data Warehousing, Metadata, And Architecture.*

1. INTRODUCTION

With the introduction of the relational data model, Data management systems were popularized in the 1970’s. Since then, these systems have evolved from being used primarily for transactional processing workloads to systems that integrate and store large amounts of data primarily for analytical purposes. These analytical systems, commonly referred to as data warehouses, support complex analysis of data and decision-making in organizations. Data warehousing facilitates the use of technologies such as on-line analytical processing (OLAP), decision support systems (DSS), and data mining software, all of which try to make sense of the large amounts of data generated by organizations [1][2][3].

The success of data warehouses in the business world motivates us to examine its use in a scientific environment. To scientists, the tasks of collecting, storing, and analyzing data are part of their core activity. Scientists are in the business of generating knowledge from data, yet, database systems in general, and data warehouses in particular, are not as popular in the scientific community as they are in the business community. One reason for this is that traditional database systems assume that the data and the processes generating the data are well defined. Scientific data and processes on the other hand are inherently shifting with domain knowledge. A data model for such an environment needs to be flexible and should easily allow such data/schema evolution without rendering historical data useless. This is especially hard to implement in traditional database systems due to structural rigidities imposed on the data types and relationships among the data. Furthermore, scientific research generates large amounts of data with complex relationships. For example, scientific laboratories conducting behavioral experiments may collect data with high dimensionality and millions of data points per experiment.

The design of a data warehouse for scientific data storage and analysis is an ambitious undertaking due to the extreme heterogeneity of the domains that supply the relevant data and the corresponding data sources and data structures themselves. In general, designing a data warehouse for supporting scientific research faces a very complex integration task. To succeed, the data warehouse design must be flexible and readily extensible, so that it can be modified to accommodate data from new domains and with new data structures as the nature of the data sources evolves over time.

The challenge from a data modeling and analysis point of view is to develop a model that captures the complexity and richness of the data while creating ancient framework for storing, querying, and analyzing scientific data.

1.1 Problem definition

One of the main objectives of this paper is to propose a data-warehousing framework, to address the problems of managing and analyzing large amounts of scientific data emerging from satellites and also generated from scientific behavioral experiments. In order to accomplish this goal, we have identified the following that are as follows:

- Identify applicable models and technologies and develop a data warehouse for a specific scientific problem.
- Develop tools or interfaces that allow researchers to query and analyze scientific data.
- Illustrate the value added by the data warehouse system in terms of facilitating more efficient management and analysis of behavioral data. (Future research proposed)

The remainder of the paper is organized as follows: In Section 2, we provide the challenges posed by scientific data. Section 3 gives a detailed description of the data warehouse schemas and tables. In Section 4, we provide a brief description of XML and Web Services technologies underlying the design of the data warehouse. In Section 5, we present a detailed discussion of the proposed architecture, its advantages based on our data warehouse architecture. We conclude in Section 6 with a summary and plans for future work.

2. BACKGROUND

The exponential growth in Science and Technology has lead to generation of larger amount of valuable data through experiments and theoretical research. Scientific & Technical data are growing in volume through basic / applied research in various fields, experiments, design &analysis, computer simulation, plant operations etc. with the complexity of data increasing at a staggering rate. But sufficient technology to handle or extract fully the latent knowledge within the data is still in the evolving stage.

The Extraction, Transformation and Loading (ETL) of different operational sources of data into the integrated staging area of the data warehouse is a fundamental component of data warehouse design. The staging area serves as a primary entry point into the data warehouse, where data is cleansed of nonessential, incorrect, inconsistent and redundant entries, then stored in a common format to enable consistent interpretation of similar data from different providers. It is generally acknowledged that the design of efficient data staging solutions is extremely difficult, requiring a considerable investment of time and resources [4]. This challenge is compounded for scientific applications, due to the heterogeneous nature of the data sources and schemas and the lack of very standardized representations.

Large amount of scientific data (say from satellites, remote sensing devices) is being poured by various heterogeneous sources. New Instruments are deployed to analyze and predict natural calamities like tsunami. These data sets are huge, going up to tera bytes. Scientists consider this as major problem of managing such large databases coming from different sources and different destinations. And there is no standard and effective way to find relevant data at a given point of time through simple queries. Scientists don’t want to devote much of their time in managing the data collected, but rather to investigate it. In corporate world data warehousing is the solution for managing huge amounts of data. It is the case with the scientific data too. But nevertheless both kinds of data (i.e. corporate and scientific) are different.

Challenges posed by the real time scientific data sets are:

- Data from heterogeneous sources.
- Complex relationships between the data types. For example, each experiment or task is a set of movements to different spatial targets. Data for each movement towards a target is stored in separate files. Each file has metadata describing global

aspects of the movement, and trial specific metadata.

- Without Meta data, the storing of scientific data is absolutely meaning less.
- Metadata in scientific community is a relatively dynamic entity. It changes as per the needs of the scientists and their field of investigation.
- Quantity of data to be handled.
- Lack of query tools: This makes data management a daunting task. For example, a simple question such as “Do we have enough rows for analysis of xyz that satisfies some conditions?” requires a researcher to manually shift through written logs of experiments and identify rows of interest.
- Uncontrolled data redundancy: Since there is no centrally accessible and shared data repository, individual researchers copy and store data relevant to their analysis on local hard drives. Such uncontrolled redundancy wastes hardware resources and makes it hard to maintain data consistency. For example, correction of corrupt data or new experimental data needs to be communicated to every potential user.

We believe that the new surveys are in full swing providing tens of Terabytes of catalog data for the astronomical community, satellites community, remote sensing community and will be providing Peta bytes of data by the end of the next decade, covering many different wavelengths and many different epochs. The volume and quality of data from these surveys will be increasing every year, and there will be an enormous pressure from the astronomical community to integrate the separate archives into a seamlessly inter-operating entity that will allow true multi-wavelength astronomy to be performed on entire classes of objects.

3. DATA-WAREHOUSING:

A Data warehouse (see Figure 1) is best described as: “ a subject oriented, integrated, non-volatile and time-variant collection of data in support of management’s decisions.”[6]

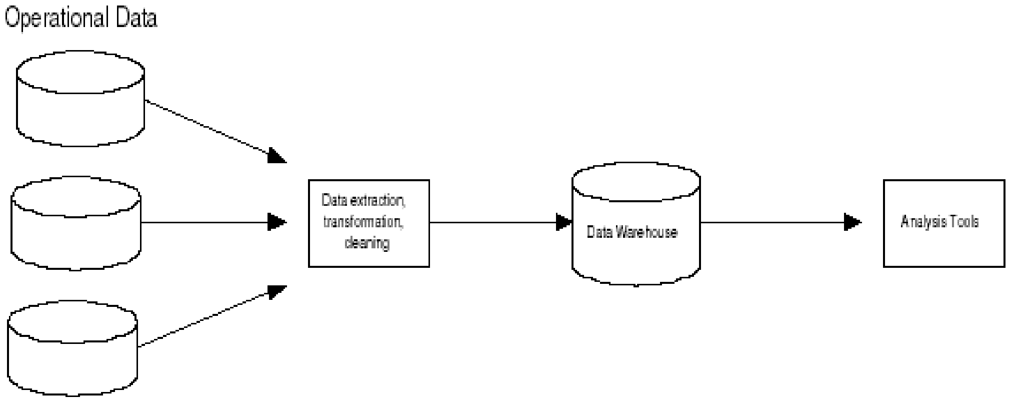


Fig. 1 A high level architecture of a data warehouse system

Subject-oriented: Data is conceptually organized by experiment metadata such as subject, task, direction, etc. The goal is to create an efficient data structure that can support the retrieval of experimental data based on metadata criteria.

Integrated: In a warehouse system, data is generally integrated from multiple operational systems. Relevant data from these systems are extracted, cleaned, parsed, and aggregated for upload.

Non-volatile: Data is stored for a long time and generally never deleted.

Time-variant: Data in a warehouse system is temporal, thus making it possible to analyze it for trends over time. Importantly, the data is temporal in the sense that the actual experimental data is collected over the period of a movement.

3.1 Dimensional/Star Schema

As defined above, data warehousing involves cleaning, aggregating and transforming source data and storing it on a platform optimized for OLAP type workloads. Our proposed schema for the data warehouse is a dimensional or star schema. The dimensional schema (Figure 2) is a simplified relational schema that minimizes the number of table joins. Krippendorf and Song describe it as: "*a central fact table or tables containing quantitative measures of a unitary or transactional nature (such as sales, shipments, holdings, or treatments) that is/are related to multiple dimensional tables which contain information used to group and constrain the facts of the fact table(s) in the course of a query*" [6].

The two key types of tables in a dimensional schema are described below:

3.1.1 Fact table: Kimball describes the "facts" in a fact table as numerical measurements of a business taken at an intersection of all dimensions.[7] Facts are generally numeric, continuously valued (not discrete), and additive. In our context, we have measurable scientific facts. The unit of measurement of the facts determines the granularity of the fact table. For instance, in our case, we can define a trial level granularity (that is, the basic unit of access would be an entire trial from an experiment), or define a fine granularity whereby each instant in time of a trial is individually accessible through any structured query language.

.

3.1.2 Dimension table: A dimension table gives identity to the facts in a fact table. As seen from Figure 2, keys derived from the dimension table identify each data point in a fact table. Dimension table attributes are generally textual and discrete [7]. For example, in Figure 2, a store dimension attribute such as location is textual (city names), and discrete (finite set of cities). The dimensional model advocates denormalized dimension tables (dimensional data is not necessarily distributed across different tables to minimize redundancies).

The reason for this is that dimension tables are relatively small compared to the Fact table, so the cost of introducing redundancy is relatively small. By avoiding normalization on the dimension table, we reduce the number of relational joins (tables joined using primary and foreign keys) in the schema, thereby improving performance for large select queries. For instance, in most cases, a large query on the fact table, will involve at most one dimension and thus one join. The dimensional schema has been widely used in data warehouse projects and is popular for business applications [8]. By minimizing the number of joins, a dimensional schema ensures optimum query performance for OLAP type workloads. Furthermore, fewer joins ensure that SQL queries are simple and do not require a deep understanding of the data model, and thus enables novice users to easily submit ad-hoc queries to the warehouse system. This is particularly valuable in our case, since the scientific community do not spend time on the management of data but rather investigation of the result out of data management

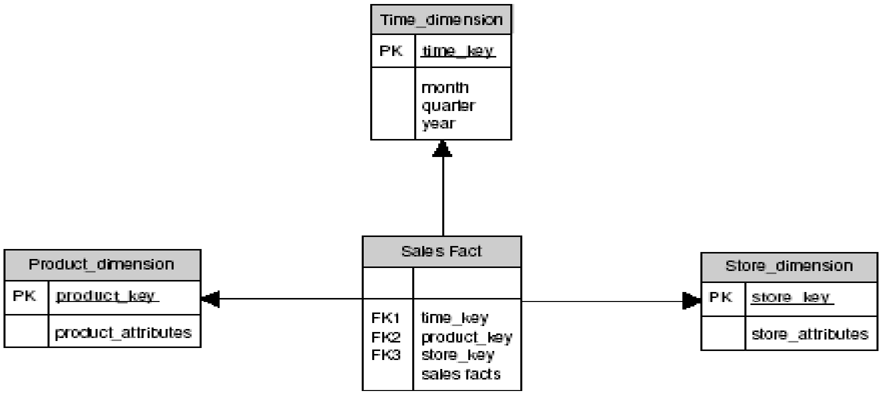


Fig 2. A sample dimension or star schema. The sales "fact" is joined to three Dimension tables, each describing a different aspect of the fact. For example, one could aggregate sales facts based on a product or a store.

4. XML AND XML BASED TECHNOLOGIES

XML was developed in 1996 to serve as a foundation for data exchange and to provide metadata markup, and has achieved rapid acceptance in both the scientific and commercial worlds. [9][10]

The proposed schema design takes the advantage of established open standards for XML and Web services. An important feature is a logical and physical shift of the data integration process away from warehouse developers and towards data providers. For scientific applications, this shift is expected to yield significantly improved data quality, since individual domain experts will be best qualified to map their respective data models to the warehouse exchange specification. Although motivated by a particular scientific application, our proposed approach is quite general and can be applied to any scientific domain amenable to XML based data exchange.

It also is becoming increasingly common for scientific funding agencies to encourage collaborating groups within a research program share data using XML [11][12]. We further progressed our idea by identifying several additional advantages of the Web services/XML architecture, for example, the reuse of existing solutions for data exchange models and the facilitation of application development requiring partial or complete integration of data from various sources.

To the best of our knowledge, the primary role played by XML in data warehouse applications to date has been in the implementation of data marts and OLAP (on-line analytical processing) cubes [13][14][15]. We have found only a few examples of its use in data ETL and integration [16][17]. The prototype data warehouse described here makes use of a full range of XML technologies to implement the scientific data ETL, integration into the staging area, and data cleansing. These include XML, XML Schema, and Web Services technologies [18][19].

The main purpose and objective of organizing data in a description-valued manner supports the natural way that scitintists process data. XML is regarded as the good and widely adopted open standard for packaging information together with a description of its meaning.

The Extensible Markup Language was introduced in 1996 in a publication by the World Wide Web Consortium (W3C) [9][10]. It immediately attracted significant attention because of its simplicity and flexibility. XML enables the presentation of information in a textual format using a markup-based approach. Each piece of data is surrounded by short labels indicating its meaning.

5. ARCHITECTURAL FRAMEWORK

First, let us consider a hypothetical scenario, though it very much exists in the real world. We prefer to call it hypothetical because of its generic nature. Say for example, there are X different sources from where data about weather are collected for a country Y. Each of these sources produces around a terabyte of data each year. During the early periods of installation of these X different sources there was very little or no emphasis placed on the archiving of these data sets. Data was stored at central location, and any one wanting to use the data will have to manually check the catalogue and then ask for a particular set of data that might be of interest to the concerned. The problem of manually looking for finding out relevant subset of data that might compound the data is useful to the task at hand.

The whole process results in a huge delay in getting the required data. Often researchers need more information than this to decide whether or not the given data is useful to the given problem i.e. more metadata for a given data is required. Once retrieved, data had to be filtered, stored, processed and the queries written should have any meaningful information from the data. All this requires a lot of time and many other resources, which otherwise could have been used to distribute to perform critical research work. Also with so many agencies being formed (specially countries in India) and several countries collaborating, same data is being archived by any/all agencies. This is again leading to wastage of resources.

If a system is developed which can be accessed by various agencies or individuals with emphasis on maintaining metadata for each data file then it will relive scientists of data management issues so that they can focus on only the research issues. We propose here an XML schema based architectural proposal(see Fig.3) to retrieve data and analyze.

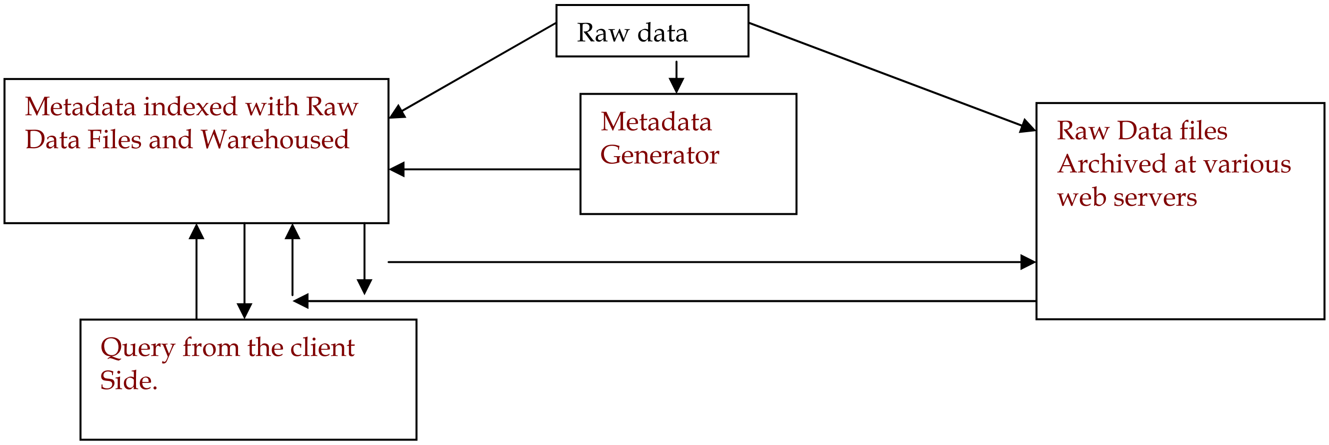

Raw data
Metadata indexed with Raw Data Files and Warehoused
Metadata Generator
Raw Data files Archived at various web servers
Query from the client Side.

Figure 3 proposed prototype architecture for storing data

This proposed architecture archives the data in real time. The architecture has two components: one that keeps the metadata and the other that keeps the raw data in its clean and reduced format. The raw data is processed through the GENERATOR. The Generator is used to create a metadata for a given data. A XML file is the output of the processed data, which contains the metadata. Metadata is stored in warehouse and a tag is attached to the metadata about the data file that it belongs to. The raw data is cleaned and stored in various web servers. A client sends a query based on the metadata to the warehouse. Based on the query the warehouse will seek all the data files that are related to the metadata from the various web servers and return them to the client. Raw data is identified with a URL. A client can directly ask for a data file if it knows the URL of the particular file. The biggest advantage with this model is that any kind of data can be processed. All we need to do is coding the GENERATOR according to a particular need. The architectural schema is presented in figure 4.

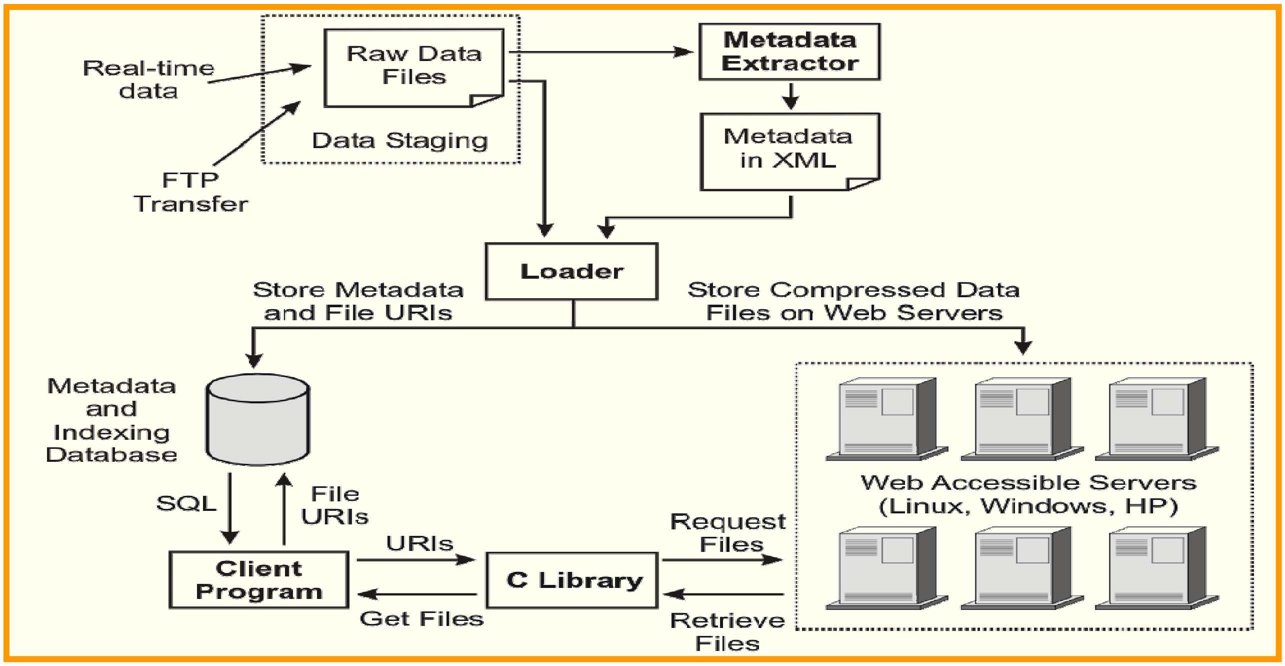
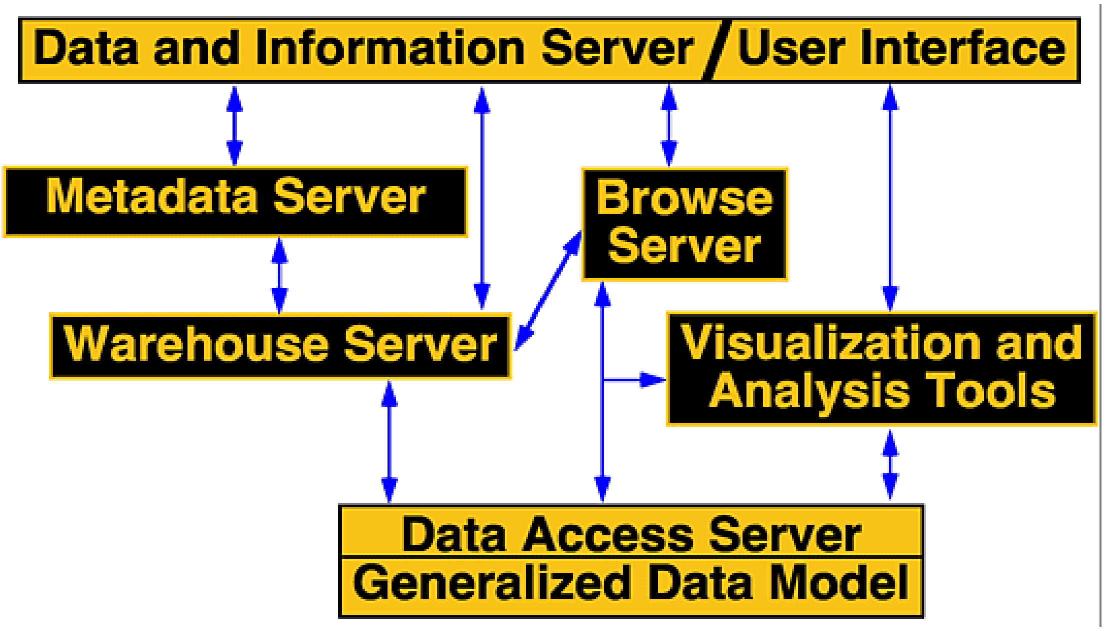


Fig 4: Architectural Schema

The design of a scientific data warehouse differs significantly from a conventional data warehouse with respect to the level of participation expected from the data providers. The data providers of a scientific date warehouse are likely to also be among the primary users of the data marts. It therefore is in their best interests to participate in an efficient data warehouse solution, even if there is some additional burden placed on them at the stage when the data is initially uploaded into the warehouse. In a conventional data warehouse, data arrives at the staging area in independent formats; it is then interpreted, transformed, and loaded into the warehouse. Thus, the integration of the data from the operational sources to the staging area is not completed until the transformed data is actually loaded into the staging area.

Fig. 5 below gives a ***generic view*** *of the above-proposed architecture.*



We have not currently proposed the concept of visualization tools. All the data sources lie in the information server. In our approach, an XML schema specifies the data exchange model to the data providers. This enables us to develop a Web services architecture where the integration of the data is completed immediately after the data has been extracted from a data sources schema and transformed into an XML document that conforms to the exchange schema. The XML document is then moved to the staging area using the Web services architecture. In other words, as a result of the Web services architecture, the integration of the data is logically and physically shifted toward the providers of the data and their source-specific implementations of extraction and transformation. Research into using XML in various capacities for data marts and Web warehousing is presently underway by others [20][21].

5.1 Expected Advantages:

The proposed architecture tries to achieve the following goals, it ties to be scalable by handling tetra bytes of a. It tries to be extensible as the type of data and metadata can change and is inexpensive as it can be made using cheap hardware and open-source software. One another advantage is that the records that have been maintained for so many years do no become a waste. It can be processed in the same manner as the new data files and can be uploaded in the warehouse. As we have already mentioned, designing the staging area and extraction, transform the component of a new data warehouse even if one takes advantage of the approach described here requires significant effort.

We have noted that although we have described a solution for our domain, and based the implementation on specific choices of software platforms, the approach itself is entirely general and can be applied to any domain facing similar design issues. Moreover, although our approach, as described, is primarily intended for a data warehousing application, it can potentially find application in other contexts requiring partial or complete data integration from diverse data sources. In future work, we are planning to continue further our investigation of the use of XML technologies in data warehousing to address other architectural components of the warehouse.

6. CONCLUSION:

Great emphasis has been laid on commercial data; it is time that scientific data got looked after. We are having an inflow of data at a rate that has never been seen as before. Hence it comes as a challenge to all to maintain this data in such a format that it remains useful and accessible by one and all. Heterogeneous data sources compound the problem of managing the data. An architecture that will use the Internet and handle large data sets is need of the hour. We mentioned the word Internet because it has become the medium through which information is being exchanged in the current century. The kind of flexibility and accessibility Internet provides is hard to ignore when we are making a new architecture. For the proposed architecture to become an applicable solution one needs to develop modules for generating Meta Data for various Research Purposes. And if one wants to add more flexibility to the system, a solution has to be found out to make this system record metadata as per the need of specific researchers, i.e. whosoever wants can archive data in his or her own way without disturbing a predefined method of storage.

REFERENCES

[1]. R. Elmasri and S. B. Navathe. Fundamentals of Database Systems. Addison- Wesley, 3rd edition, 2000.

[2]. [31] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. The Data Warehouse Lifecycle Toolkit. John Wiley & Sons, Inc., 1998.

[3]. P. Gray and H. J. Watson. Present and future directions in data warehousing. SIGMIS Database, 29(3): 83–90, 1998.

[1]. P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. Conceptual modeling for ETL processes. In: *Proceedings of the ACM Third International Workshop on Data Warehousing and OLAP (DOLAP)*, pp. 14-21 (DOLAP .02, McLean, VA, 2002).

[2]. W. H. Inmon. Building the Data Warehouse. Wiley Computer Publishing, 2nd edition, 1996.

[3]. M. Krippendorf and I. Song. *The translation of star schema into entity relationship diagrams*. Proceedings of the 8th International Conference and Workshop on Database and Expert Systems Application (DEXA) Workshop, pages 390–395, 1997.

[4]. R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. The Data Warehouse Lifecycle Toolkit. John Wiley & Sons, Inc., 1998.

[5]. P. Gray and C. Israel. The data warehouse industry. Web, February 1999. http://www.crito.uci.edu/itr/publications/pdf/datawarehouse.pdf. Current as of 14 April 2004.

[6]. World Wide Web Consortium (W3C), Extensible Markup Language, W3C Working Draft, 14 November 1996. http://www.w3.org/TR/WD-xml-961114.

[7]. World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 6 October 2000. http://www.w3c.org/TR/2000/REC-xml-20001006.

[8]. Ecological Metadata Language Specification. See: http://knb.ecoinformatics.org/software/eml/.

[9]. Data Specifications for The National Cancer Institute Director’s Challenge. http://dc.nci.nih.gov/tools/DataManagement.

[10]. N. T. Binh, A. M. Tjoa, and O. Mangisengi. MetaCube-X: An XML metadata foundation for interoperability search among Web warehouses. In: *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW 2001)* D. Theodoratos, J. Hammer, M. Jeusfeld, and M. Staudt, Eds., Interlaken, Switzerland, June, 2001.

[11]. R. M. Bruckner, T. W. Ling, O. Mangisengi, and A. M. Tjoa. A framework for a multidimensional OLAP model using topic maps. In: *Proceedings of the Second International Conference*

on Web Information Systems Engineering (WISE 2001), Web Semantics Workshop, 2:109-118 (IEEE Computer Society Press, Kyoto, Japan, 2001).

[12]. M. R. Jensen, T. H. Møller, and T. B. Pedersen. Specifying OLAP cubes on XML Data. Technical Report 01-5003, Department of Computer Science, Aalborg University (2001).

[13]. B. Ensink, K. Haveman, T. Schavey, and M. Shrestha. XML based adaptation of the composite approach for database integration. In: *Proceedings of the 37th Annual ACM Southeastern Conference (CDROM).* (ACM Press, New York, NY, 1999).

[14]. Z. G. Ives, A. Y. Halevy, and D. S. Weld. An XML query engine for network-bound data. The VLDB Journal 11(4): 380-402 (2002).

[15]. World Wide Web Consortium (W3C). XML Schema. See: http://www.w3.org/XML/Schema.

[16]. World Wide Web Consortium (W3C). Web Services Architecture. See: http://www.w3.org /TR/ws-arch/.

[17]. D. Pedersen, K. Riis, and T. B. Pedersen. XML-extended OLAP querying. Technical Report 01-5003, Department of Computer Science, Aalborg University (2001).

[18]. O. Mangisengi, J. Huber, C. Hawel, and W. Essmayr. *A Framework for Supporting Interoperability of Data Warehouse Islands Using XML*. (Springer-Verlag, New York, 2001).