Data Warehousing and Data Extraction on the World Wide Web

Aparna S. Varde
Computer Systems Engineer
Lucent Technologies, MA, USA.

Abstract

The unprecedented volumes of data today existing in a variety of places and formats make it imperative to have some techniques for data integration. The growing popularity of the Internet places an additional demand of making information quickly and easily available on the Web for wider access. This opens a window to the world of Web Database Integration. In Lucent Technologies' Optical Networking Group (ONG) at Merrimack Valley (MV), we have developed applications in this field. We have explored Data Warehousing Concepts which emphasize the use of integrated data stores called *warehouses*, optimized for distribution, and data structures called *marts*, optimized for access to support end-user analysis of data. The focus of this effort has been Capacity Planning and Forecasting in the Circuit Pack Engineering Organization. In ONG, we have also delved in to the area of Data Extraction using XML and Web.sql. This encompasses Web-querying techniques for underlying databases and dumping relational database tables to provide Web views catered to the end-user. The prime application here has been in the area of Maintenance and Calibration Control Systems (MACCS) in the Work Services Group at the MV Plant. There are also related applications of Extraction concepts in the Circuit Pack Engineering Organization. A summary of our research and development in the Web Database world is presented in this paper.

1. Introduction

There are vast amounts of data all over an organization, stored by different people at different times. Data could exist anywhere and in any format such as flat files, relational tables, object-oriented databases, HTML pages and so forth. The advent of the Internet has led to wider access and this means more data to manage. This places an even greater demand on an organization to provide quick and easy access to information as and when needed. So, raw data is scattered all over the place and the need is to get all the information in a single store in a meaningful format, for fast and simplified access to anyone at any time for any purpose. It is this need that led to the concept of Data Warehousing. As the very name implies, a Data Warehouse is a huge store of information in an integrated form. It facilitates distribution of data to end-users who can in turn use it for decision-making.

There are several definitions of a Data Warehouse [1], but the following is most widely accepted: "*A Data Warehouse is a subject-oriented, time-varying, non-volatile collection of data used primarily in organizational decision-making*." Each of the terms used here deserves some explanation. A warehouse is *subject-oriented* because all the information stored there is primarily focused on one main subject related to one organization e.g. in our case all the data chiefly deals with circuit pack manufacturing. The term *integrated* is appropriate since the individual information sources that fed the warehouse may have data stored in their own respective formats, but when the warehouse is being built, useful pieces of this information are gathered in one universally accepted format for storage. Thus all the data is not simply transferred from source to warehouse, but integrated as per the requirements. It is *time-varying* data because every piece of it was originally stored in a source at a specific time and bears the appropriate time stamp, and this source data may vary over a period due to modifications, updates and the like. However, it is *non-volatile* meaning that once it is stored in the warehouse, it stays there just the way it was initially stored. It does not change automatically with corresponding changes in the underlying information sources. For this reason, a warehouse needs to be refreshed from time to time. Updates need to be incorporated and outdated information needs to be purged. In Data Warehousing, the focus is primarily on high-level decision-making, rather than on day-to-day information used in transaction processing in regular databases.

2. Applications

Some of the main applications of Data Warehousing are summarized below.

- *Helps in DSS and EIS systems*: The realm of Management Information Systems in the industry today presents several arenas. Among them are Decision Support Systems [2] and Executive Information Systems that provide help at the managerial level and above in what-if scenario analysis and making key decisions for the company. Having all the information in one integrated store for easy access greatly aids such systems.
- *Goes hand-in-hand with ERP*: Enterprise Resource Planning and Material Resource Planning [3] are extremely important for the present and the future, especially in product-based organizations. Data Warehousing provides for better planning by providing all the information upfront in order to build ERP tools.
- *Enables more effective Data Mining*: Data Mining [4,5] involves digging useful pieces of information from sources, analogous to digging out ores from a mine. One can build association rules, useful output relations and so forth that make the user's job easier in pulling out information. Again it can be clearly seen that a warehouse would certainly enhance the mining process, by providing a single huge store of meaningful information.
- *Gives overall integrated approach to information*: The buzzword in organizations today is "information". And this is precisely why Data Warehousing is so popular. It gives integrated, timely and meaningful information to engineers, managers and external users, thereby pulling the whole organization together to function in a more sophisticated manner.

3. Technology in Warehousing

We are aware that traditional databases have been using the OLTP technology, i.e. On Line Transaction Processing. This primarily involves the automation of day-to-day data processing applications. The operations in OLTP are atomic, repetitive and detailed. The chief performance criteria here are data consistency and transaction throughput. The systems resulting from OLTP are basically the TPS or Transaction Processing Systems.

Data Warehouses, on the other hand emphasize more on integration and decision support, as is evident from the applications mentioned above. So the focus is more on overall consolidated information available at a glance, rather than on the specific details of individual transactions. It is like visualizing the forest, rather than the trees. So the performance metrics here would be related more to query throughput and response times. The technology well-suited to this is OLAP, i.e. On Line Analytical Processing. This deals more with data consolidation, and complex analysis of information. This is used in Warehouse Creation [6]. Some interesting OLAP tasks are:

- *Roll-up:* This involves going to a higher level of aggregation in the integration process, i.e. reducing detail
- *Drill-down:* Conversely, this goes down an aggregation level, implying an increase in detail.
- *Slice and Dice:* This is analogous to the select and project operations in regular databases.
- *Pivot:* This involves re-inclination of the data view.

So, we can see that the operations here are more high-level. In order to implement OLAP technology, the original model used is ROLAP (Relational OLAP). This uses the simple RDBMS model, where data is stored in tuples, and bears attributes. However, in warehousing, there is also a trend towards Multidimensional OLAP or MOLAP. This adds more dimensions to store information, so we are not restricted to 2D-tuples. We now have more sophisticated data structures for example 3D data cubes, complex arrays and more. The emphasis in MOLAP is on multiple facts and multiple dimensions.

4. Life Cycle of a Data Warehouse

A warehouse is an extensive structure involving several phases of development. A fully developed warehousing system typically has the following parts.

• *Information sources*: These are the building blocks for the warehouse. They are the original sources of the data like flat files, RDBMS, OODBMS and others. The raw data exists here and needs to be integrated into the final warehouse.
• *Wrappers*: Each information source has a module called a wrapper [7] that is primarily responsible for data transformations prior to the actual integration of data. The functions of a wrapper during warehouse creation include data gathering from the sources, data cleansing, and format conversions. Once the data is available in a generic format required by the warehouse, the consolidation can be done. After the warehouse is up and running, there are periodic updates needed to refresh the information. The wrapper is also responsible for the tasks involved at the time of these updates.
• *Integrator:* The warehouse on the whole has one integrator module. This filters, summarizes and merges the data from the individual wrappers and finally dumps the desired information into the warehouse.
• *Warehouse:* Needless to say, this component is the actual store of integrated information.

These are the essential components of a warehouse. The Data Warehousing Life Cycle includes Warehouse Creation and Warehouse Maintenance. The detailed steps involved are

1. *Requirements Analysis:* This is similar to the initial step in any software life cycle model, where facts are collected, needs are outlined and detailed specifications are written up accordingly, so as to serve as a guideline for development.
2. *Architecture:* A Data Warehouse could be centralized or distributed. It may require a dedicated server. These logistics are worked out and the servers and tools are selected.
3. *Data Modeling:* In order to logically analyze raw data, we need to construct data models. These help us identify the entities in the system, their relationships with each other, their properties, common features and the like. Data modeling includes terminology like E-R diagrams, star and snowflake schemas, materialized views and more. Going into a greater level of detail with the methods and terminology would make us delve into the area of Data Modeling, a realm by itself.
4. *Layout:* Once the logical analysis is done, we need to proceed with the actual layout. We need to identify the individual information sources, apply ODBC connections and continue from there.
5. *Meta-Data:* This is "data about data". A simple example would be a library where the books form the data and the catalog forms the meta-data giving information about the books i.e. about the data. In a Data Warehouse, meta-data is crucial, because the focus is on analysis rather than transactions. So we need to identify a meta-data repository, find a suitable location for it (centralized, distributed etc. depending on warehouse architecture) and proceed to build access mechanisms for it.
6. *Extract, Transform and Load:* These are the three basic steps involved in physical data transfer. The data is extracted from the underlying information sources in a raw form, transformed to the desired format by the wrapper module and then loaded to the warehouse by the integrator module in a consolidated manner.
7. *Monitoring and Administration:* The above steps all dealt with warehouse creation. Once the warehouse is built, we have to keep it up and running. We need to set up support for this. We have DWAs (Data Warehouse Administrators) analogous to DBAs (Data Base Administrators). We have to develop mechanisms for fault detection and correction, recovery from breakdown and the like in order to ensure reliability of functioning.
8. *User Interface:* A warehousing system could have a simple character interface but this would make access quite cumbersome defeating the very purpose of a warehouse namely, quick at-a-glance information for analysis and decision-making. So considerable amount of time and effort is spent on building an interface to cater to the needs of the user.
9. *View Maintenance:* As the underlying information changes, the updates need to be reflected in the warehouse [8]. In regular databases, this happens automatically, but in a warehouse, the updates need to be physically transferred. We could refresh the warehouse in batches at certain intervals. During such updates the warehouse itself ceases to be functional. The system is shutdown, updates are carried out, and then it is

up and running again. On the other hand, the system could reflect the changes as and when they occur, without causing a system shutdown. This has to be done one transaction at a time. The former approach is called batch updates and the latter is incremental updates. Each has its own pros and cons. More on this when we discuss research issues.

10. *Purging:* Once the warehouse data gets old, it has to be driven out to make room for new fresh information. Also we need the latest up-to-date information for correct analysis and decision-making. This process of removing the old data that is of no use, is called purging. It is essential for effective functioning of the warehouse.

5. Data Mart

A Data Warehouse life cycle is rather extensive. The whole development process can be very time consuming and involve tremendous cost and effort. Since building a complete warehouse can get complicated, organizations sometimes like to begin with developing a smaller system called a Data Mart.

A Data Mart can be defined as "*a departmental subset of a Data Warehouse focused on selected subjects*" [9]. While a Data Warehouse is optimized for distribution, a Data Mart is optimized for access. This is analogous to a warehouse and a mart in common parlance.

A warehouse is usually used for storage and distribution of goods while individual marts or shops can draw specific categories of items from the warehouse and make them readily available to the general public. Similarly a Data Warehouse acts as a huge integrated data store while a mart serves as a smaller entity that focuses on a particular area and is well suited for end-user access. Thus a Data Mart is simpler to build and maintain.

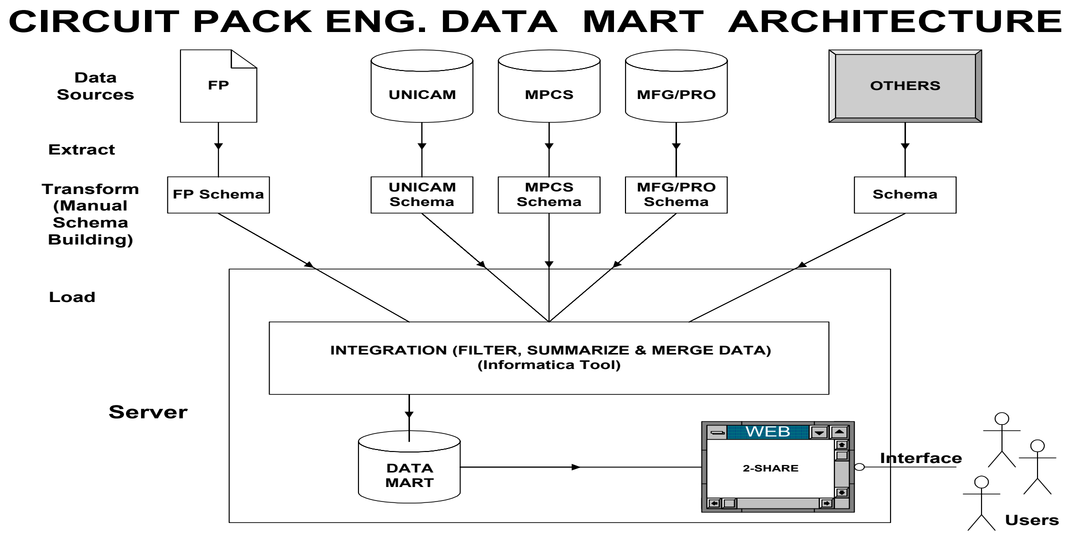
A common strategy is to develop a Data Mart first, and use it as a pilot for a larger project to build a complete warehouse. If the mart functions well, serving the desired purpose of information analysis and decision-making effectively, then we can move further in that direction and work towards building a warehouse. This approach provides for better cost-benefit analysis.

Sometimes a Data Mart may be the prime requirement of the organization, and they may not want to consolidate the system into a full-fledged warehouse. This could happen in the case where the environment itself is relatively smaller and a mart would be more appropriate to cater to the needs of the users.

Our organization is building a Data Mart with a focus on Capacity Planning in Circuit Pack Manufacturing. The raw data involved here is primarily from legacy systems in our circuit pack engineering organization. Most of it is from RDBMS sources, but some of it is also present in the form of UNIX files, hard paper copies or just general information scattered around the place. The Data Mart organizes this systematically and makes it available to the end-user through a user-friendly Web-interface.

The mart includes an analysis tool that allows the user to do some forecasting, what-if scenarios and decision-making in the area of capacity planning. The targeted users are mainly the managerial and higher executive level employees who need summarized information at a glance without going into the details of engineering aspects or operational tasks. The architecture of our Circuit Pack Engineering Data Mart is shown in Figure 1.

Figure 1: CPE Data Mart



6. Web Warehousing

Data Warehousing concepts are being applied over the World Wide Web today. Traditionally, simple search engines have been used to retrieve information from the Web. These serve the basic purpose of data recovery, but have several drawbacks. Most of these engines are based on keyword searches limited to string matching only. That narrows down our retrieval options. Also we have links, at times several levels of them in a particular context. But simple search engines do not do much justice to obtaining information present in these links. They provide direct information recovery, but not enough indirect link information. Also if we have files related to certain subjects and need to couple these, the coupling has to be done manually. Web search engines do not provide mechanisms to incorporate the above features. These and other reasons have led to further research in the area of Web knowledge discovery and have opened the window to the world of Web Warehousing [10].

There has been increasing development in query mechanisms [11] over the World Wide Web. We have languages such as WebSQL, Web3QS, WebLog, Web3QL and the like. These languages are a combination of database query techniques that are SQL-based and Web retrieval methods that have pattern-matching searches and user-friendly formatted displays. Thus, a user can for example, query over the Web-based on a search criteria like "circuit packs manufactured before jan 99 mounted on double sided boards meant for the 400G family of products." Such information would be hard to get using simple keyword-based search engines. A query like this can be executed using SQL in an RDBMS for a specific database. However, this would require knowledge of SQL, RDBMS packages, the underlying system and more. With Web Warehousing, we can obtain responses to these and much more complicated queries, using just simple English commands, and these queries can be spread over a wider system than a single database. Moreover they display the output in a format easily readable by a user who may not be familiar with any programming language. So they are a step towards DSS (Decision Support Systems).

Web warehousing incorporates the Web Information Coupling Model (WICM). This has Web nodes and links, Web schemas and tuples, Web SPJ (select project join) operations and other concepts thar are analogous to the corresponding concepts in regular databases. We are also able to do Web Ranking of Hot Tuples. This implies ranking specific tuples as most wanted, least used etc. based on criteria such as frequency of occurrence, place of occurrence and so on, in order to facilitate retrieval. That in turn leads on to yet another area of Mining over the Web, again analogous to regular data mining. We can build structures called Web Bags with a focus on certain subjects. There is also a terminology called Warehouse Concept Marts similar to Data Marts optimized for access. More details on Web Warehousing are beyond the scope of this paper. However Web warehousing has tremendous potential and poses several areas for further research and development.

7. Data Extraction over the World Wide Web

The data extraction mechanisms [12] that were applied to regular databases can now be used for querying information over the World Wide Web. Queries such as the ones described above under Web Warehousing can be applied to get specific results that meet the needs of the user. These could be in terms of Web views, textual responses and more.

There is a growing interest in the field of XML, the eXtended Markup Language. XML is the universal format for structured documents and data on the Web. It allows the user to alter the form of the information without going into the details of its contents. So the user need not have knowledge about how an underlying data source is built and maintained.

Consider for example, that we display circuit pack information by family of packs, and within each pack display details like the code, manufacturing date, components etc. We then wish to alter the display, so that we get a serial ordering by code rather than by family. This can easily be done using XML without going into the language of the underlying information source. These tasks if executed using XML just require one simple statement from the user of the type "order by family" or "order by code". The user need not fathom the design of the data source, or send a query in a language that the source understands, or write a piece of code for the same. XML makes things much easier.

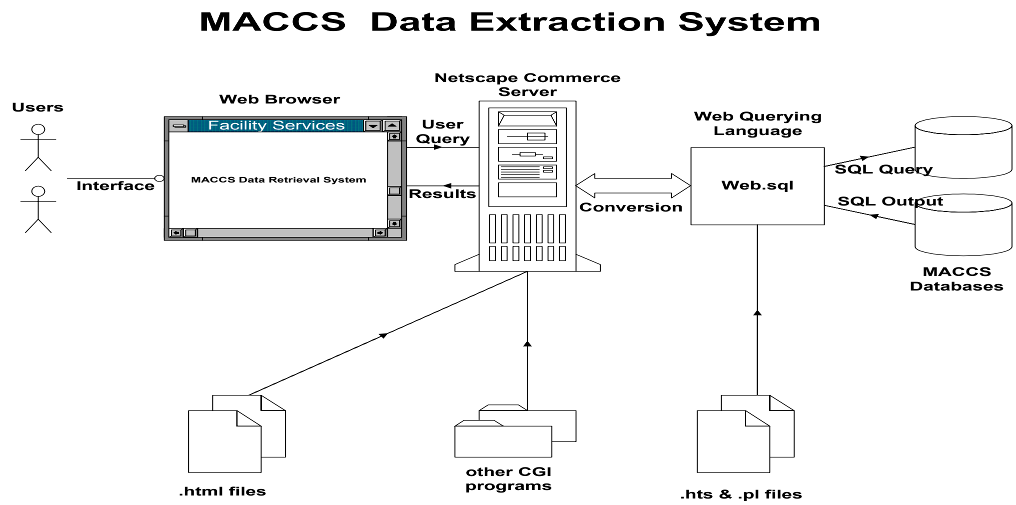
There are XML products being built and used in industry. One example is the 2Share product built by 2Bridge Software. We used this in the Circuit Pack Manufacturing Data Mart and in some other areas like our Engineering CAM projects. This proved a useful tool for Data Extraction over the Web. In the CPE Data Mart Architecture, it can be seen that this XML-based tool has been used at the interface level in the mart for user-friendly access.

Another interesting product is Web.sql developed by Sybase. This provides a library of pro-functions for database access to the end-user. It embeds the procedural language Perl and query language SQL directly in HTML, with Javascript to create forms. We used this in the Maintenance and Calibration Control System (MACCS) in the Work Services Group at ONG. It enabled us to extract information about Factory Engineering Job Requests (FEJRs) and Facility Requisitions directly over the Web using plain English-like commands.

The earlier extraction mechanisms used UNIX and required the end-user to have knowledge of shell programming, other UNIX commands and SQL. This meant that the upper management and customers always required engineering support to obtain even the simplest of information from MACCS.

With the new system built using Web.sql, the access is greatly simplified and users can execute complicated queries like "FEJRs requested by a particular person and designed for a certain product and altered last on a specific date." we can continue to add more search criteria at the simple point and click of a mouse, directly over the Web, without looking into the organization and UNIX interface of MACCS. The architecture of the MACCS Web Data Extraction System is shown in Figure 2.

Figure 2: Maintenance and Calibration Control Web Data Extraction System



8. Research Issues

Since Data Warehousing is by itself a relatively new technology, there are many areas yet to be explored. There are several loose ends in the various phases of the warehousing life cycle. Each of them poses research issues. Broadly classified, the research areas of Data Warehousing fall under the categories of warehouse creation, warehouse maintenance and data extraction. We can consider each one separately.

8.1 Warehouse Creation

The data from the information sources needs to be cleansed [4]. For example, if we obtain the data from a table that has missing values or wrong values, these cannot be dumped directly into the integrator. We could carry out error checks, delete columns with missing values or insert default values in them. Any of these options would in turn pose related problems. Likewise there are problems associated with the translation of input data, i.e. the format conversion and representation of information. There are cases of structural and semantic heterogeneity.

Structural heterogeneity refers to differences in field sizes, scales and units, which is self-explanatory. Semantic heterogeneity refers to issues like synonyms, homonyms and the like. For example salary and income although two different names may actually refer to the same entity, while an abbreviation like PIM may refer to Product Integration and Manufacturing in one place and Parts In Master in the other.

There could be several offshoots of issues such as these, and dealing with them at the wrapper level would involve mechanisms coded into the design of the wrapper. We need to do Entity Identification at some point and study the relationships between them. The problem described above, associated with entities, can also have its counterpart in attributes of these entities. Sorting that out is known as Attribute Value Conflict Resolution. Moving from the wrapper to the integrator, we have other warehouse creation concerns.

The algorithms that are used for data integration at this level need to be designed such that extensibility becomes easy, because a warehouse needs to cater to future expansion, which will be affected at the integrator. So scalability of algorithms is a prime concern. Needless to say the efficiency is a key issue as well, particularly when we look at the cost-benefit aspects of a warehouse.

Optimization and improvement are constantly underway. Yet another research area is that of more interactive tools for schema integration that would ease the burden on the developer. Once we have made satisfactory progress in most of these avenues, we need some metrics for accuracy of information in the warehouse v/s the time spent on integrating that information. However, going into metrics and measurements seems rather far-fetched at this point. It could become a hot subject of future research work in warehouse creation. The other areas mentioned above provide considerable scope for research today [6]

8.2 Warehouse Maintenance

Once a warehouse is created, it needs to be maintained. This maintenance and upkeep itself poses a whole set of issues. We know that the data in the underlying information sources keeps changing and this change is not automatically reflected in the warehouse, unlike ordinary databases. So we need to work on a methodology to reflect these updates in the data sources at the warehouse level.

Updates could be either batch updates or incremental updates. *Batch updates* are done at regular intervals, determined by the nature of the application. The interval could be a day, a week, a month, a quarter or even a year. During the batch update sessions, a warehouse is shutdown and is not available to end-users for access. Once the refreshing is done and the latest information is consolidated into the warehouse, then it is made available for regular operation. Clearly one drawback of this scheme is that it is inefficient due to periods of system shutdown. Also the information that the end-user accesses during the interval following the occurrence of an update, but preceding its batch consolidation into the warehouse, is obviously not the most up-to-date information and this in turn can lead to more inconsistency issues. However, the advantage of batch updating is that it is a relatively less expensive approach. The other mechanism is *incremental updates*. Here, the update that occurs at the source is integrated into the warehouse almost immediately after it occurs. So we do the updates, one at a time rather than in batches. This certainly overcomes the two problems of the batch approach namely system shutdown and information inconsistency. But the implementation of incremental updates is far more complicated and hence involves greater cost and more research in the actual methodology and techniques. There are other approaches, such as self-maintenance and update filtering.

The *self-maintenance* approach keeps an auxiliary (individual) view of each information source at the integrator level, rather than a materialized (combined) view of all the sources. So the changes in each source can go directly to the integrator through the auxiliary view, making things easier. But this introduces the problem of increased space complexity, since the storage of so many auxiliary views would occupy too much space in the system.

The other method called *update filtering* is the technique of incorporating some intelligence into the information sources, so that they selectively filter out only those updates that are of consequence to the warehouse from an access point of view. The remaining updates are not even propagated to the integrator. This implies a more complicated architecture of the wrappers for each source. So we see that each approach for refreshing solves some problems but poses new ones, and hence opens up more areas for research. Warehouse maintenance issues are mainly related to updates.

In case of Web Warehousing there are more security issues involved [13]. Other concerns are the management of meta-data, fragmentation of the database in case of a distributed warehouse, permanent removal of outdated information, and the development of user-friendly tools that enable better Data Warehouse Administration. Again, each of these areas in turn would lead to more specific problems and one can go deeper into investigating and coming up with solutions.

8.3 Data Extraction

Once a Data Warehouse is built and is being maintained, the extraction of information also becomes a key issue [12]. The most common extraction mechanisms is sending queries to the warehouse and obtaining the results. If we do Web warehousing, the sending of queries becomes relatively easier due to a user-friendly interface. In either case, the queries need to be optimized, so that we do not have redundancy and make efficient use of resources. This query optimization poses a research area by itself.

We also need to look at building views catered to the end-user and redefining those views with respect to changing user needs. Another factor is the application of domain knowledge. When this knowledge of the real world is incorporated into query processing, in addition to the information contained in the warehouse, we are likely to get more practical results. But introducing real world knowledge in the system implies building a certain amount of intelligence into it, and this in turn opens up a whole new set of issues.

Yet another progressive field of research deals with parallelism in query processing. We can incorporate Select-Project-Join operations into the queries spread over several information sources, particularly those queries that are frequently encountered, and then preserve the results in the form of knowledge bags as described in Web warehousing terminology.

This leads on to the area of Web mining and incorporates supervised and unsupervised learning. Local and global information coupling can be done across the Web, since we come across many related links that could be preserved together. Finally, we could even incorporate some DSS features into extraction. If the decision support functionality is built into the warehouse, then we are making our systems truly intelligent, making things much simpler for the users. That is the ultimate goal of a good Data Warehouse --- to enable effective decision-making. It is clearly seen that each of the topics opened up here can branch on to a range of subtopics. Ideas are invited for further development in the respective areas.

In a nutshell, there is substantial scope for research in the area of Data Warehousing. Most of the current research in the Data Warehousing world is focused on query processing and view maintenance issues. The future is promising.

9. Summary and Conclusions

Data integration is crucial in today's world. This has led to the development of Data Warehousing. Data Warehousing is a hot area presently in the corporate sector, as well as in academics and research. OLAP technology is being used in current applications and is being further explored for more advancement. Web Warehousing and extraction opens a whole new world of research and development.

From a Lucent ONG perspective [14], Data Warehousing will benefit the manufacturing organization that has parts and products spread all over numerous locations, and need integration. There are warehouses and marts up and running in different departments. Our examples are the CPE Data Mart focusing on Capacity Planning, and the MACCS Web Data Extraction System focusing on Factory Engineering [15]. The introduction of warehousing and its Web applications [10,13] are benefiting internal and external users.

Further research from our side is ongoing, especially in the area of warehouse creation [6], inconsistency issues and the like. We look forward to more from the R&D standpoint, and hope to brainstorm for new ideas and implement more challenging projects in the world of Data Warehousing and extraction.

Acknowledgements

The author would like to thank Edward March, Michael Schrowang, William Batchelder, Anthony Cappabianca, Ricky Li, Thomas Hallworth, Richard Vaillancourt, Lewis Liu from Lucent Technologies, and Elke Rundensteiner from Worcester Polytechnic Institute, for their contributions.

References

[1]. S. Chaudhari, U. Dayal,
An Overview of Data Warehousing and OLAP Technology, VLDB Conference - 1996

[2]. S. Bansal
Real World Requirements for Decision Support - Implications for RDBMS, SIGMOD - 1995

[3]. E. Geiger
ERP and Data Warehousing Meet, DB2 Magazine - Summer 1999

[4]. W. Kim,
I/O Problems in Preparing Data for Data Warehousing and Data Mining, Part 1, JOOP - February 1999

[5]. W.Kim,
Data Warehousing and Data Mining, Part 2, JOOP - May 1999

[6]. J.Shrivastava, P. Chen,
Warehouse Creation - A Potential Roadblock to Data Warehousing, IEEECS - Oct 1998

[7]. M. Roth, P. Schwarz,
Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, VLDB Conference - 1997

[8]. D. Agrawal, A. El Abbadi, A. Singh, T. Yurek,
Efficient View Maintenance at Data Warehouses, ACM-1997

[9]. K. Duncan, J. Thomann, D. Wells, W. McKnight, D. Marco, M. Haisten, D. Wells
Proceedings of the Fourth Annual Implementation Conference, TDWI - February 1999

[10]. J. Gruser, L. Raschid, M. Vidal, L. Bright
Wrapper Generation for Web Accessible Data Sources,
International Conference on Cooperative Information Systems - 1998

[11]. V. Vassalos, Y. Papakonstantinou,
Describing and Using the Query Capabilities of Heterogenous Sources, VLDB Conference - 1997

[12]. C. Squire,
Data Extraction and Transformation for the Data Warehouse, SIGMOD – 1995

[13]. R. Kimball
Data Webhouse Security Threats, TDWI - February 1999

[14]. D. Mog, F. Grimes, R. Alferness
Bell Labs Technical Journal on Optical Networking, BLTJ - January-March 1999

[15]. B. Domanski
Distributed Capacity Planning, Parts 1&2, Enterprise Systems Journal - January & February 1999.