Data Warehousing - An End-To-End Solution

Howard Ong

Principal Consultant

Aurora Consulting Pty Ltd

Abstract

One of the greatest misconceptions of Data Warehousing today is the notion that a Data Warehouse is simply a strategic deployment of one or more tools. Strategic it may be, but a Data Warehouse is not just about tools. It is first and foremost an architecture. This paper introduces the concept of End-To-End Data Warehousing, which focuses on establishing a framework of concepts centred on the Data Warehouse architecture. It examines key human and technological factors that influence the success rate of the Data Warehouse. It explains the various components of the Data Warehouse architecture, their functions and interrelationships. Finally, the paper reviews the repertoire of Data Warehouse tools available today - from relational to multidimensional; from ROLAP to MOLAP. The paper explains how these tools and technology can be optimally deployed in a Data Warehouse architecture, as either off-the-shelf or custom-built solutions, delivering the right information to the right user at the right time.

About The Author

A frequent presenter at local and regional conferences, Howard has been a proficient user of database technology since 1991. In particular, Howard possesses in-depth experience in the planning and development of Business Intelligence and E-Business Applications. Harbouring a keen interest in the application of advanced database in these areas, Howard and his team have helped many organisations deploy BI and E-Business solutions using Oracle, Microsoft and IBM technology. Since 1997, Aurora Consulting has been delivering Data Warehouses and E-Business solutions to a wide variety of organisations including large government departments and ASX-listed companies.

For more information on Howard or Aurora Consulting, visit http://www.aurora-consult.com.au or email info@aurora-consult.com.au.

Introduction

With the barrage of Data Warehousing tools in the market today, one can be forgiven for thinking that a Data Warehouse is simply a strategic deployment of tools. This is perhaps the most commonly misunderstood of all Data Warehousing concepts today. This paper aims to dispel this common misconception and proposes a new way by which Data Warehousing should be viewed - from an end-to-end perspective.

End-To-End Data Warehousing

The End-To-End Data Warehousing proposed in this paper is *not* a development methodology - there are enough of those around already ! Rather, it is a perspective by which a project manager should adopt in developing a Data Warehouse. From the author's own development experience, an end-to-end perspective maintained throughout the development project and beyond, would greatly enhance the chances of success and the long-term viability of the Data Warehouse.

The Top End

At the top end of the perspective, it is imperative for a Data Warehouse project manager to maintain focus on a number of critical success factors throughout the project. These critical success factors can be classified into 2 broad categories :

- Human Factors
- Technological Factors

Like all strategic systems, it is important for a Data Warehouse to achieve the synergistic fusion of these 2 elements : *human* and *technology*.

Human Factors

Sponsors And Champions. The project sponsor and the project champion provide the vital link between the Data Warehouse and the user community at large. Some typical characteristics of the project sponsor and champion are :

- Ideally, the project sponsor should be a medium- to top-level manager from a non-IT discipline. The project champion, on the other hand, is likely to be an IT-literate knowledge worker reporting to the project sponsor.
- Both the sponsor and champion are likely to be direct stakeholders of the initial Data Warehouse. The initial subject area is likely to be under the sponsor's direct responsibility; while the initial Data Warehouse should assist the project champion in performing his or her routine duties.
- Unlike most other systems within an organisation, a Data Warehouse cuts through every facet of an organisation's business activities - from sales activities in the marketplace, to manufacturing processes on the factory floor, to cost accounting at the head office. For such a new concept to take root in an organisation, much promotion and education is required. To this end, the project sponsor and champion serve as vital allies to the Data Warehouse project team from within the ranks of the user community.

At this point, it is worth clarifying that while the discussion above assumes that the project sponsor and champion are 2 separate individuals, this is not a hard-and-fast rule. On one extreme, the project sponsor and the project champion can be the same person; on the other extreme, the project sponsor and champion can be represented by a small community of stakeholders. The identity of the project sponsors and champions can also change, as Data Warehouse development progresses from one subject area to another. Where the sponsors and champions comprise of a group of stakeholders, it is important that potentially conflicting aspirations and expectations of the different stakeholders be skilfully managed. Otherwise, a group of allies can easily turned into a group of enemies !

The Project Team. Data Warehouses are a new breed of system with significant differences from traditional application systems - from database configuration to the development life cycle, paradigm shifts abound. One of the greatest challenge of a Data Warehouse project manager is undoubtedly the seeking and enlisting of appropriately skilled individuals for the project team. As experienced Data Warehouse developers are in short-supply, it is often necessary to compromise experience for attitude. In the absence of relevant experience, a willingness to shift one's development paradigm is a valuable asset to the project.

Project Funding. Data Warehouses are expensive to build and is associated to lengthy development cycle. However, being a new concept to the organisation, the Data Warehouse project often faces the paradoxical challenge of having limited funding to achieve an initial quick-win prototype. Further funding for the project usually hinges on the success of this initial prototype. In another paper entitled "The Evolution Of A Data Warehouse Architecture - One Size Fit All ?" (Oracle Open World Singapore 1999), the author explained a number of evolutionary routes by which the quick-win requisite can be achieved, without over-compromising the architectural goal of the Data Warehouse.

Technological Factors

ETL Considerations. In a nutshell, Extraction, Transformation and Loading (ETL) activities transforms *operational data* from the source systems into *analysis data* in the Data Warehouse. Examples of ETL activities include data loading, data merging, conversion, validation, derivation and summarisation. These activities may sound straight-forward and unsophisticated, but consider this : the repertoire of source systems that contribute data to a Data Warehouse is likely to vary from standalone Excel spreadsheets to mainframe-based systems many decades old. At the point of conception, most of these systems were *never* designed to be integrated with each other - at least not at the level required by the Data Warehouse. Non-standard data coding, differences in unit of measures, multiple systems using the same name for different data items, multiple systems using different names for the same data item … and the list goes on. In particular, where the same data item (or derivatives thereof) exists in more than one system - one should not be surprised that their contents do not agree with each other ! In fact, most experienced Data Warehouse developers would agree that ETL is the single most important and time-consuming activity of the entire development process - it has the potential to make or break the project.

While a description of the steps involved in formulating a sound ETL strategy is well beyond the scope of this paper, the author will nevertheless like to highlight the following considerations for the readers to ponder over :

- What data items are required and from which source systems ?
- What are the data reconciliation activities required to integrate these data items ?
- How frequent should each data item be loaded ?
- Are there any new data item that can be derived from the existing ones ?
- In what order should be the ETL activities be carried out ?
- Where one ETL activity is ordered after another, which sort of dependency relationship exists between them ? If the former fails should the latter go ahead ?

Data Granularity. Data granularity is the level of detail or summarisation of data residing in the Data Warehouse. It directly determines the volume of data loaded and stored in the Data Warehouse. Higher volume of data is usually associated with more voluminous extraction, more frequent data loading, greater need for disk space, more complex summary management and more elaborate tuning of the database. Hence data granularity has a direct bearing on the level of complexity in the development and administration of the Data Warehouse. Therefore, Data Warehouse developers should strive towards storing and manipulating data at as high a level of granularity as possible. However, the author would like to emphasise that one should never lose sight of the users' requirements in the quest for higher level of data granularity. Where loading and storage of detailed data is inevitable, architectural components such as Operational Data Stores (ODS); storage strategies such as data retention period; and summarisation techniques such as rolling summary should be employed to improve the manageability of the Data Warehouse.

Evolutionary Path. At the onset of this paper, the author emphasised that a Data Warehouse is not a collection of tools - it is an architecture. In a paper entitled "The Evolution Of A Data Warehouse Architecture - One Size Fit All ?" (Oracle Open World Singapore 1999), the author highlighted a number of evolutionary paths by which a matured Data Warehouse architecture can be evolved. The Data Warehouse Evolutionary Plan should be formulated early in the Data Warehouse project and be refined and improved as the project progresses. This Evolutionary Plan constitutes a strategic plan of the Data Warehouse project. To be without it is like embarking on a journey without a navigational map.

Corporate Data Model. At full maturity, a Data Warehouse schema is a physical manifestation of an organisation's Corporate Data Model. Just as the

Evolutionary Plan is likened to a navigational map, the Corporate Data Model is likened to the destination - both are essential components to a smooth journey. As the Data Warehouse is built incrementally, it is important to drive each development effort from the Corporate Data Model. This helps to ensure that any new subject area can be integrated into the existing Data Warehouse jigsaw. This technique is most useful where an up-to-date Corporate Data Model exists prior to the Data Warehouse being developed. However, it is not uncommon for an organisation to be without a Corporate Data Model - let alone an up-to-date one. In this situation, the Data Warehouse becomes a de facto, evolving Corporate Data Model. Even under such circumstances, the technique highlighted does not change - new subject areas should always be built *integrated* into the existing Data Warehouse schema.

Often, when analysis needs arise for new subject areas, political, costing and time pressure may tempt the Data Warehouse developer into developing standalone Data Marts rather than expend the extra effort to integrate with the Corporate Data Warehouse. As tempting as it is, such a stop-gap strategy is myopic. Over time, as the organisational landscape is proliferated by standalone, non-integrated Data Marts, supported by a web of non-standardised extraction processes, the strategy will cost the organisation greatly in terms of credibility and productivity.

The Red Centre

The choice of appropriate software, though important, does not guarantee successful Data Warehouses. In fact, a Data Warehouse is not one, or even a combination of software products - it is an architecture of system components. This *Data Warehouse Architecture* forms the centre of any Data Warehouse project.

Described below are some common components of a Data Warehouse Architecture and their interrelationship. Note that not all of these are essential components to a Data Warehouse architecture. In fact, even the infrastructure by which they are brought together may vary from warehouses to warehouses.

- Operational Source Systems. These are operational systems that supply the Data Warehouse with data. These systems can range from decade-old legacy systems to small standalone Access databases.
- Operational Data Store (ODS). The ODS is a repository of analysis data of a fine granularity and typically of a low retention period. It is commonly used in scenarios such as that of a retail outlet chain, where the Sales Director would like to track sales figures on a daily basis. However there is

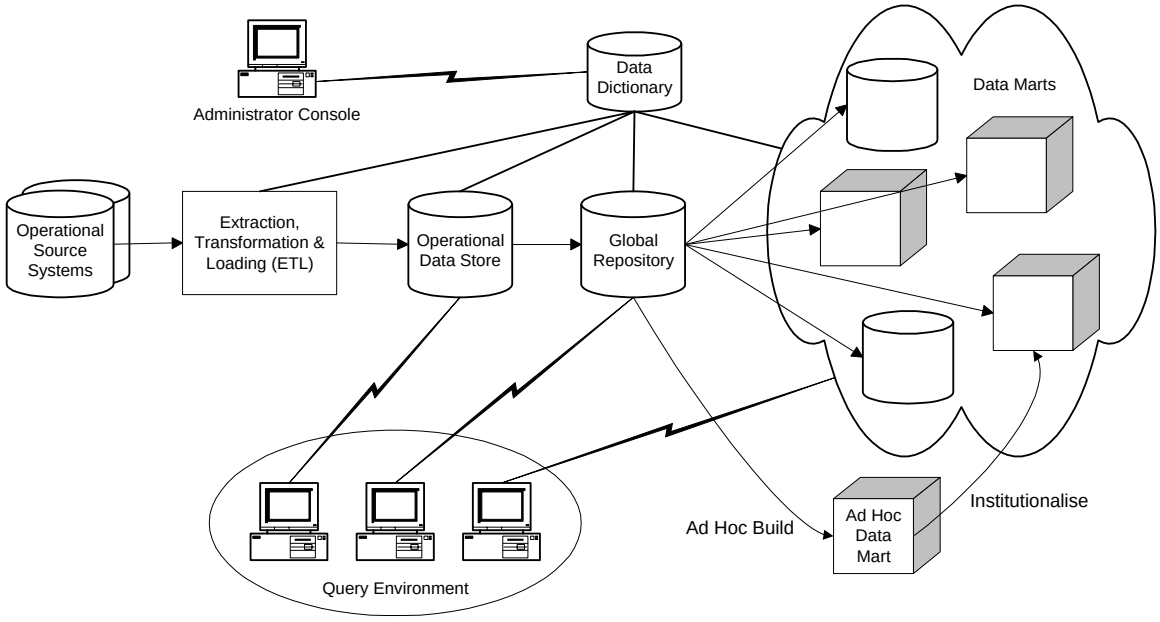

Figure 1 A Matured Data Warehouse Architecture

usually little likelihood of such fine granularity data to be required over a long period of time. For example, the Sales Director may wish to compare the current day's sales figure with that of last month or even last quarter, but probably not beyond. ODS' can play a key role in keeping the total size of the Data Warehouse manageable.

- Extraction, Transformation and Loading (ETL) processes. The ETL processes transform and load operational data from the source systems into analysis data. It is equivalent to what Microsoft called *Data Transformation Services (DTS)* and what Oracle called *Extraction, Transportation and Transformation (ETT)*. Common functions of the ETL processes include data loading, manual data entry, data merging, conversion, validation, derivation and summarisation.

- Global Repository. Often considered the "Data Warehouse" itself, the Global Repository is a repository of all Data Warehouse data. It can probably be described as the heart of the Data Warehouse, serving as both a convergent and a divergent point for all ETL activities. At the risk of further confusing the semantics of this new emerging discipline, the author coined the term "Global Repository" to distinguish it from the "Data Warehouse", which is in fact not a repository but the parent architecture. The Global Repository is typically a ROLAP or a 3NF relational database.

- Data Dictionary. If the Global Repository is the heart of the Data Warehouse, then the Data Dictionary is surely the brain of the Data Warehouse. The Data Dictionary houses the Data Warehouse's meta data. Two types of meta data may reside in the Data Dictionary : structural information such as measure dimensionality and dimension structures; and ETL process information such as derivation logic, validation procedures, status of data loading processes, etc.

- Administrator Console. The Administrator Console performs Data Warehouse administration tasks such as maintenance of dimensions and measures; definition of ETL processes, monitor ETL runs, etc.

- Data Marts. Data Marts house subject-specific subsets of information from the Global Repository. Depending on the requirements, these Data Marts can either be implemented using ROLAP or MOLAP technology. Some advantages of Data Marts over the Global Repository are simpler database structure, versatility of user interface, improved performance, richer functionality, and in some cases, portability. From time to time, additional Data Marts can be built on an ad hoc basis to analyse particular subject areas of interest. If desired, ad hoc Data Marts can be institutionalised and become part of the pool of permanent Data Marts.

- Query Environment. Depending on the needs, a wide variety of query tools can be deployed in the query environment. These can be traditional client-server tools (Oracle Reports, Crystal Reports); web-query tools (Oracle PL/SQL Cartridge, Cold Fusion); custom-built applications; ROLAP tools (Oracle Discoverer, Cognos Impromptu); or MOLAP tools (Oracle Express Objects, Cognos PowerPlay).

Note that not all of the above-mentioned components are present in all Data Warehouses. In fact, a bare minimum Data Warehouse would only have the Operational Source Systems, ETL Processes, some form of data repository and a Query Environment. However, some components like the Global Repository and the Data Dictionary, though not essential, are highly desirable.

The Bottom End

The choice of appropriate technology and tools is an important critical success factor to the Data Warehouse project. The choice of the right Data Warehouse tool is as important to the Data Warehouse as the choice of the right database and query tools for a traditional OLTP Application. The preceding section describes how the Data Warehouse Architecture is made up of several components. The diversity in the Data Warehouse Architecture gives rise to a wide variety of Data Warehousing tools in the market today. This section explains the various categories of Data Warehousing tools and how the various software tools fit into them. However, with the large number of tools available, any attempt at evaluation and explanation of their features would warrant a separate paper (if not a book !). Instead, this paper concentrates on query-related tools, with only a brief mention of other Data Warehousing tools.

Relational Databases

Contrary to some beliefs, relational databases have a major role to play in a Data Warehouse architecture. Not only is it an ideal technology for the Data Dictionary and the Global Repository, it is also the basis for a new breed query tools called ROLAP. In Data Dictionaries and Global Repositories, the relational database is employed very much like any traditional OLTP application - with a number of notable exceptions in terms of database optimisation, constraint definition and backup requirements. In ROLAP applications, however, special adaptation is necessary - both in terms of database features and database design. Many relational database vendors, such as Oracle and IBM, have added additional features to their respective relational database products to better support ROLAP design such as star and snowflake schemas. In addition, relational databases dedicated to Data Warehousing have also emerged. One such database is Informix's Red Brick Warehouse.

ROLAP Tools

Relational OLAP tools run against relational databases, adapting traditional relational technology for OLAP analysis. The greatest advantage of these tools is their ability to deliver quick-win OLAP prototypes by leveraging off an organisation's investment in relational technology. Thus, compared to the other OLAP query tools, ROLAP tools enjoy cost-savings in terms of licenses and human resources. However, limitations in the current breed of ROLAP tools means that the user interface and system performance are likely to be less impressive compared to other OLAP tools. Some examples of ROLAP tools are Oracle Discoverer, Cognos Impromptu and Aurora's proprietary DataView product.

MOLAP Tools

OLAP analysis, by its very nature, is multidimensional. Multidimensional OLAP tools perform OLAP analysis against bona fide multidimensional databases. Some MOLAP vendors bundle the MOLAP query tool and the multidimensional database as a single product suite. Others license the tool and database separately. Typically, MOLAP tools enjoy superior user interface and impressive rapid development capabilities. Thus, these tools have the advantage of gaining quick converts by delivering graphically impressive results within a short span of time. Some examples of MOLAP tools are Oracle Express, SAS MDDB and Cognos PowerPlay / Powercube.

Hybrid OLAP Tools

Hybrid OLAP tools deliver OLAP analysis by employing both relational and multidimensional technology. There are 2 major ways by which Hybrid OLAP analysis is delivered.

- The multidimensional caching approach creates a (usually small) multidimensional cache by querying a relational database. This multidimensional cache then serves out query results in a manner not unlike most MOLAP query tools. Where necessary, relational data are re-queried and refreshed. Business Objects' Microcube and Corvu's Dynacube are examples of such caching technology.
- The drill-through approach supports independent multidimensional and relational databases, each holding summarised and elementary data respectively. Its fundamental assumption is the age-old 80-20 rule - that the data in the multidimensional database is sufficient to satisfy most queries most of the time. Where more detailed data are required relational data are fetched via a *drill-through* operation from the multidimensional database. Microsoft SQLServer OLAP Services (Plato) is one example of such technology.

It is worth noting that the lines between ROLAP, MOLAP and Hybrid OLAP have somewhat blurred in recent times. For example, the Relational Access Manager and better Express-Discoverer integration has provided Oracle OLAP users 2 separate means by which drill-through can be achieved. Cognos, on the other hand, has long since supported drill-through capability from PowerPlay to Impromptu.

Custom-Built Query Tools

With the Global Repository usually being a traditional relational database, one cannot discount the value of good-old relational reporting tools such as Oracle Reports, Crystal Reports and Cognos Impromptu. Often, the best way to win

converts is to simply provide them with non-fanciful relational reports from an integrated repository of warehouse data. The very sight of data from different Operational Source Systems appearing in a single report is usually enough to impress new users.

Other Tools

The discussion thus far has concentrated on query tools. In the present climate of tool proliferation, many other types of Data Warehousing tools exist and more are emerging everyday ! Some of these are :

- Data Mart tools. These tools provide a rapid development environment for the definition and management of simple ETL processes. Some examples are : Oracle Warehouse Builder, Oracle Data Mart Suite, Ardent Data Stage, Prism and Cognos Decision Stream.
- Data Dictionary. Currently, the market for Data Dictionary tools is a somewhat disjoint one. Most OLAP and Data Mart tools are shipped with its own dedicated standalone Data Dictionaries. Increasingly, CASE tools such as ERWin and Oracle Designer are being used to capture the design of Data Warehouses - making these repositories Data Dictionaries in their own rights. Thankfully, convergence and standardisation is on the way. Independently, Oracle and Cognos have both been working on developing common meta data repositories for their respective products. Other companies have introduced independent Data Dictionary products with a broad focus, rather than narrowly focusing on supporting a front-end OLAP or Data Mart product. One such tool is Aurora's proprietary DataWare product.
- Data Mining tools. OLAP tools are most useful where there are preconceived relationship between business measures - they *prove* these pre-existing concepts by a theory deduction process. Data Mining tools is a new suite of tools working in the opposite way. These tools interrogate existing data in order to *create* new concepts on relationship between business measures - via a theory induction process. Some examples of Data Mining tools are : Oracle Darwin, Cognos Scenario and Cognos 4Thought.

Conclusion

Data Warehouse being the new IT discipline that it is, will continue to enjoy the explosion of ideas and advances it is currently experiencing. Academic concepts will continue to be refined and new technology will continue to emerge. This paper aims to contribute the advances of this fast exploding field by providing a new perspective by which a Data Warehouse project should be planned and managed. While this paper suggests a number of pointers and guidelines, it should be emphasised that there is no one silver bullet to successful Data Warehousing. It is imperative that organisations assess their specific circumstances and needs. Where possible, organisations should seek the advice of Data Warehouse specialists to provide expert guidance as it embarked on its Data Warehouse evolutionary route. Remember this golden rule : a well-planned journey is more likely to give you a smoother ride !