

**AN OLAP-BASED PM PROTOTYPE FOR COST CONTROL AND  
MANPOWER ALLOCATION**

by

HAO (HOWARD) NIE

B.Sc., Southeast University, 1999

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Civil Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

October 2005

© Hao (Howard) Nie, 2005

## **Abstract**

In the past decades great efforts have been made in applying Information Technology (IT) in the construction industry. The existing IT-integrated PM tools seem to be helpful to improve the efficiency of a single project control task, such as scheduling, but they lack a level of integration and coordination that can help manage construction projects from multiple perspectives in a unified manner. Although the OLAP (On Line Analytical Processing) technology has been widely used for business management, known as “Business Intelligence” (BI), it has not been well adopted by the construction industry in the project management domain.

The main objective of this thesis research is to test and evaluate whether the fast growing OLAP technology can facilitate project management in the construction industry, at least in some selected aspects. A Web-based, OLAP-integrated PM prototype was established to do the testing and evaluation. A data warehouse with multidimensional data structure was established in Microsoft Access, followed by the design and processing of OLAP cubes through Microsoft SQL Sever Analysis Services. The Microsoft BI Portal was utilized as the front-end OLAP solution for multidimensional data representation and analysis. Two OLAP cubes were developed in the prototype for project cost control and manpower allocation analysis respectively, and the testing and evaluation was carried out by combining the OLAP features with several realistic project management scenarios.

The thesis concludes that the OLAP technology does offer potential improvements to project management, but a great deal of future work is needed to extend the prototype to a well-rounded system. The benefits of this OLAP-base PM prototype, along with its shortcomings and limitations, are summarized at the end of the thesis. Also included is the discussion and suggestion of the future research work in this area.

## Table of Contents

<b>Abstract .....</b>	<b>ii</b>
<b>Table of Contents .....</b>	<b>iii</b>
<b>List of Tables .....</b>	<b>vi</b>
<b>List of Figures .....</b>	<b>vii</b>
<b>Acknowledgement .....</b>	<b>ix</b>
<b>Chapter 1: Introduction to the Research .....</b>	<b>1</b>
1.1. Research Background.....	1
1.2. Research Motivation.....	1
1.2.1. Industry Needs .....	1
1.2.2. Technology Opportunities.....	5
1.2.3. Summary of Research Motivation .....	6
1.3. Research Objective.....	6
1.4. Research Scope.....	6
1.5. Prototype Features.....	7
<b>Chapter 2: Overview of OLAP Technology .....</b>	<b>8</b>
2.1. Introduction .....	8
2.2. Overview of OLAP and Data Warehouse.....	8
2.3. Key Concepts in OLAP and Data Warehouse .....	11
2.3.1. Dimensions, Levels, and Members .....	11
2.3.2. Measures and Aggregation.....	13
2.3.3. Data Cubes .....	14
2.3.4. MDX & PivotTable Services .....	15
2.4. OLAP Database and Multidimensional Data Analysis .....	16
2.4.1. Cube Structure and OLAP Database.....	17
2.4.2. Slice & Dice.....	19
2.4.3. Drill up & Drill down .....	19
2.5. Office Web Components (OWC) .....	20
<b>Chapter 3: Project Management &amp; OLAP Technology .....</b>	<b>23</b>
3.1. Introduction .....	23
3.2. Related Research .....	23

3.2.1. Research on Integrating Construction Information.....	23
3.2.2. Research on Multidimensional Data Modelling for Construction Information.....	24
3.3. Overview of Mapping OLAP to Project Management.....	25
3.4. Dimensions.....	27
3.4.1. Time Dimension.....	28
3.4.2. Activity Dimension.....	30
3.4.3. Participant Dimension.....	31
3.4.4. Product Dimension.....	32
3.5. Measures and Aggregations.....	33
<b>Chapter 4: A Proposed OLAP-based PM System .....</b>	<b>37</b>
4.1. Introduction .....	37
4.2. Overview of an OLAP-based PM Platform .....	37
4.3. Simplifications of the Proposed Prototype.....	38
4.3.1. Data Storage.....	39
4.3.2. Number of Dimensions.....	41
4.3.3. Type of Measures.....	41
4.3.4. Data Presentation.....	42
4.3.5. Cube Structure.....	42
4.4. Functionality & Benefits of the Proposed Prototype.....	43
4.4.1. OLAP Integration.....	43
4.4.2. Web-based Presentation.....	43
4.4.3. Friendly UI.....	43
4.4.4. Structure Consistency.....	44
4.5. Sample Project.....	44
<b>Chapter 5: Data Warehouse Model and OLAP Cube Design.....</b>	<b>47</b>
5.1. Introduction .....	47
5.2. Data Warehouse Design.....	47
5.2.1. Dimensional Model Design.....	47
5.2.2. Fact Table Design.....	49
5.2.3. Dimension Table Design.....	50
5.2.4. Multidimensional Structure.....	52
5.2.5. Sample Project Data.....	53
5.3. OLAP Cube Design & Pre-processing.....	54
5.3.1. Overview.....	54
5.3.2. Set Up Data Source Connections.....	54
5.3.3. Build OLAP Data Cubes.....	56
<b>Chapter 6: Web-Based Data Presentation .....</b>	<b>60</b>
6.1. Introduction .....	60

6.2. Microsoft .NET and Web Services .....	60
6.3. Microsoft Business Intelligence Platform .....	61
6.3.1. Overview.....	61
6.3.2. Microsoft SQL Server 2000.....	62
6.3.3. Microsoft Office System.....	63
6.4. Microsoft BI Portal.....	65
6.4.1. Overview.....	65
6.4.2. BI Portal Architecture .....	66
6.4.3. BI Portal Installation and Set-up.....	67
<b>Chapter 7: Prototype Testing and Evaluation .....</b>	<b>69</b>
7.1. Introduction .....	69
7.2. Pre-testing Preparation .....	70
7.3. Testing & Evaluation – Project Cost Control .....	73
7.3.1. Overview.....	73
7.3.2. Cost Data Drill Through .....	73
7.3.3. Cost Data Slice.....	78
7.3.4. Cost Data Slice & Dice .....	84
7.3.5. Advanced Cost Data Manipulation .....	89
7.4. Manpower Allocation .....	93
7.4.1. Overview.....	93
7.4.2. Scenario 8 - Manpower Allocation over Time.....	93
7.4.3. Comments on OLAP-based Manpower Allocation .....	96
<b>Chapter 8: Conclusion and Future Work Trends .....</b>	<b>97</b>
8.1. Introduction .....	97
8.2. OLAP-based PM System versus Conventional PM Systems.....	97
8.3. Thesis Conclusion .....	98
8.4. Future Work.....	101
<b>BIBLIOGRAPHY .....</b>	<b>103</b>

## **List of Tables**

Table 2.1: A sample cube cell from the “Retail” cube .....	18
Table 2.2: Summary of OWC Components [Microsoft, 16].....	21
Table 5.1: OLTP Operational Database vs. Data Warehouse Database [Browning, 14].....	48

## List of Figures

Figure 1.1: Schematic of the Dimensions in a Unified Approach to PM [Froese, 2] .....	4
Figure 2.1: Worldwide Total OLAP Market .....	9
Figure 2.2: Business Intelligence (BI) with OLAP & Data Warehouse [Microsoft, 5] .....	10
Figure 2.3: Traditional two-dimensional data structure .....	12
Figure 2.4: A Typical “Time” Dimension [Microsoft, 9].....	12
Figure 2.5: Concept Model of the “Retail” Data Cube.....	14
Figure 2.6: A “Star” Schema of the “Retail” Cube .....	17
Figure 2.7: A “Snowflake” Schema of the “Retail” Cube .....	18
Figure 3.1: The Project Life Cycle of a Constructed Facility [Hendrickson, 1].....	29
Figure 3.2: Traditional Construction Participants.....	32
Figure 4.1: Data Flow Chart for a Typical OLAP Solution .....	40
Figure 4.2: Simplified Data Flow Chart for the Proposed Prototype.....	41
Figure 5.1: Data Warehouse Star Schema Diagram.....	49
Figure 5.2: Data Warehouse Fact Table Structure .....	50
Figure 5.3: Data Warehouse Structure – “Star” Schema.....	53
Figure 5.4: Set Up System Data Source Name.....	55
Figure 5.5: Make Data Source Connection.....	56
Figure 5.6: Define Cube Measures .....	57
Figure 5.7: Define Cube Dimensions .....	57
Figure 5.8: Processing Data Cubes in MS SQL Server Analysis Services .....	58
Figure 5.9: Summary of “Cost Control” Data Cube.....	59
Figure 6.1: Microsoft .NET Framework Architecture [Microsoft, 22].....	61
Figure 6.2: The Microsoft® Platform for Business Intelligence [Microsoft, 25].....	62
Figure 6.3: Microsoft BI Portal User Interface.....	66
Figure 6.4: Microsoft BI Portal Architecture [Microsoft, 30] .....	67
Figure 7.1: Main UI of Microsoft BI Portal .....	70
Figure 7.2: Create New Data Sources for Cost Control and Manpower Allocation .....	71
Figure 7.3: Link Data Source for Cost Control .....	72
Figure 7.4: Link Data Source for Manpower Allocation .....	72
Figure 7.5: Cost Control – Data Overview.....	74
Figure 7.6: Drill Down “Tasks” Dimension – Level one.....	75
Figure 7.7: Drill Down “Tasks” Dimension – Level two .....	76
Figure 7.8: Drill Down “Participant” Dimension .....	76
Figure 7.9: Drill Down “Product” Dimension.....	77
Figure 7.10: Cost Info against Cost Types before “Slicing” by Time.....	79
Figure 7.11: Cost Info against Cost Types after “Slicing” by Time.....	80
Figure 7.12: Cost Info against Participants after slicing by “Time” .....	81
Figure 7.13: Cost Info against Tasks after slicing by “Products” – Overview.....	82
Figure 7.14: Cost Info against Tasks after slicing by “Products” – Details.....	82

Figure 7.15: Cost Report – “Time” vs. “Cost Type” .....	85
Figure 7.16: Cost Report – “Tasks” vs. “Cost Type” .....	85
Figure 7.17: Cost Report – “Product” vs. “Tasks” vs. “Cost Type” .....	86
Figure 7.18: Cost Report – “Product” vs. “Tasks” vs. “Time” .....	87
Figure 7.19: Multi-dimensional Cost Report with All Dimensions Involved.....	88
Figure 7.20: Add New Calculation for BCWP .....	91
Figure 7.21: Add New Calculation for ACWP .....	91
Figure 7.22: “S” Curve of BCWP and ACWP.....	92
Figure 7.23: Manpower Allocation over Time (Typical View).....	94
Figure 7.24: Manpower Allocation over Time (User-specified View) .....	94
Figure 7.25: Manpower Allocation of Each Participant Category.....	95
Figure 7.26: Manpower Allocation of Each Participant .....	96

## Acknowledgement

I would like to express my sincere gratitude to the following people who made it possible to complete this thesis research.

My thankfulness goes to **Dr. Thomas Froese**, my supervisor, whose consistent guidance and encouragement made this research project possible. Dr. Froese spent valuable time to give me advices and assistance in this research, and provided all necessary materials and software. I appreciate all his time and effort in making this thesis research completed successfully. I would also make a grateful acknowledgement for his kindness in helping me with many personal issues, making it possible for me to concentrate on my research work.

My thankfulness goes to **Dr. Sheryl Staub-French**, my co-supervisor, whose guidance, encouragement, and suggestions greatly motivated me to do my research. Her essential comments made the thesis research more thorough and comprehensive. I am also thankful for all her encouragement and suggestions in those discussions with her relating to general research matter.

My thankfulness goes to **Dr. Alan Russell**, whose academic lectures and reports greatly helped me to broaden my knowledge in construction management.

I would also like to extend my sincere thanks to all my family members who keep supporting me from all aspects during my research.

Sincere thanks to all the faculty, staff, and graduate fellows in the Project & Construction Management group that have helped me during my graduate study at UBC.

## **CHAPTER 1: INTRODUCTION TO THE RESEARCH**

### **1.1. Research Background**

Project management in the construction industry is a tough job, no matter the size and type of the project. A construction project typically involves a huge web of components that must be coordinated during the life cycle of the project. Those components may be the various participants including the owner, engineers, PM professionals, prime contractors, subcontractors, suppliers, and so on. They may also be distinct technical processes like planning, designing, financing, constructing and operating, which can be further divided into smaller tasks and subtasks. Or, they may even be different materials and products. These components are not isolated. They are always interdependent with each other, and to a large extent it is the interdependencies that make the process of construction management complex.

In the past decades great efforts have been made in applying Information Technology (IT) in the construction industry, helping to deal with the complexity of AEC/FM projects and improve the efficiency of the whole project management practice. Until recently, however, IT-integrated project management tools and technologies have shown minimal impact on project management practice in the construction industry. Computer-integrated tools may be helpful to a single task, scheduling or estimating for example, but they lack a level of integration and coordination dealing with various project data categories that can help manage projects from multi-perspectives in a unified manner.

### **1.2. Research Motivation**

#### **1.2.1. Industry Needs**

In short, the industry need for a unified multi-dimensional project management system can be explained from two aspects. One is the need to improve the presentation of “data” itself; the other is the need to improve the presentation of “information” behind data.

Although these two aspects are discussed separately in this section, they are actually highly interactive with no distinct boundary to separate one from the other.

- Project Data Presentation

On one hand, the construction industry produces a huge volume of quantitative data but so far we are lacking an efficient tool to record, store, organize, present and analyze those project data. Take cost control as an example. Cost data for a construction project is cumulative and can be voluminous as the project progresses, especially for large projects. However, in current practice, the most common cost management process relies on the transmission of Windows-based Excel spreadsheets. Staff from some organization produces cost estimates or detailed cost reports, which are passed to and reviewed by a higher level person. He or she summarizes and aggregates all received cost reports and send them to another higher level. The aggregation process repeats and continues until a final cost report is created.

The conventional practice has many problems that affect cost tracking and control. First, the process of creating report is time-consuming. At the time when the cost reports are finally created, the cost data may have already been too out-of-date to be utilized for analysis. Secondly, the report format is generally fixed and once the final cost reports are created, they are very hard to change unless the final user wants to re-organized and re-aggregate the raw data again from the beginning. The cost reporting method is thus very inflexible leading to a poor cost analysis capability. Moreover, a traditional cost report, to some extend, is just a “book” of cost data, with no efficient data visualization to provide information for cost analysis. Users have to bury themselves in the raw data to extract reasonable information.

In terms of cost management, it is time for the industry to have a system that

greatly reduces the amount of time to produce cost reports, has flexible, dynamic, and multi-view data representation formats, and provides efficient data visualization helping the managers explain and understand various aspects of project status. As for other quantitative project data that requires cognition, they follow the same rule and have the same problems.

- Implicit Information Presentation

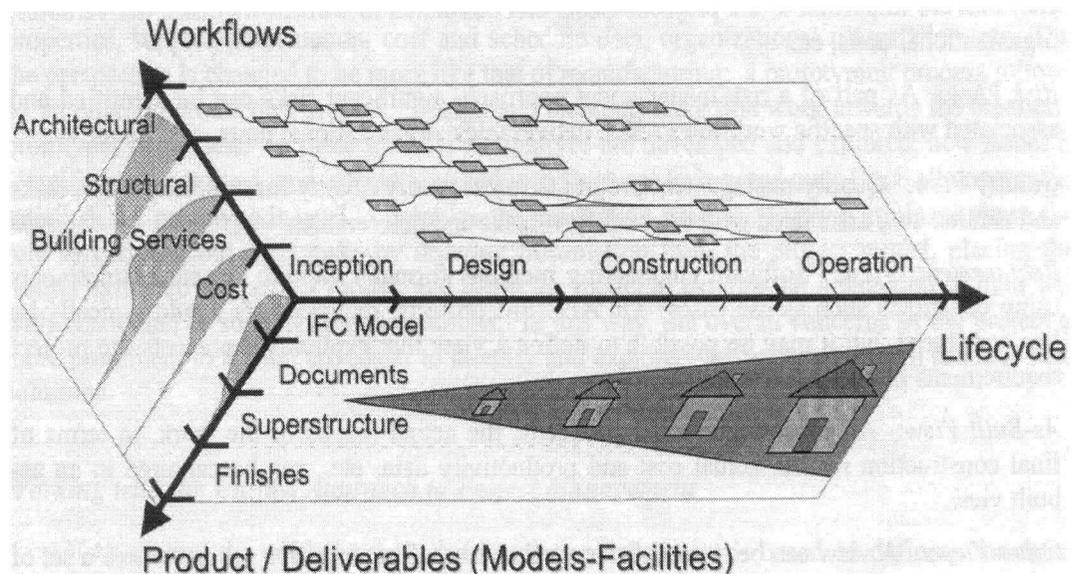
On the other hand, no quantitative data produced in the construction industry are isolated. Without linking with some external referential information, pure quantitative data are meaningless. The current project management tools, however, lack an efficient way to deliver and represent the implicit information behind the data.

Again, take cost control as an example. A single cost value is just a number that has no meaning itself; only when it is linked to some other information does it makes sense. The information can be time, tasks, participants, products and so on. If each of the information categories is regarded as a specific project view, then the quantitative cost data can be possibly distributed and aggregated over each view. In that way, users get a chance to better investigate and understand the project cost structure from a multi-view perspective.

In real-world practice however, such a multi-view of cost structure is missing, because cost reports are generally aggregated and created in a single way and do not clearly indicate linkages with other project views. Those reports provide sufficient numeric cost data but do not deliver efficiently communicated information that affects the cost data. For example, tasks (schedule) information and cost reports are usually created by different software and kept separately. As a result, project managers usually have to manually synchronize and integrate a comprehensive view from the different reports by themselves. They are often

forced to infer the cost impacts of task changes or participant changes, rather than being provided with some automatic indication to assist this process [Hendrickson, 1].

In November 2003, a unified project management (UPM) approach was introduced in the 4<sup>th</sup> Joint Symposium on Information Technology in Civil Engineering [Froese, 2]. The fact was pointed out that the existing project management practices underemphasize the interdependencies between various categories of project information. To address this weakness, the newly introduced UPM approach intends to focus on an explicit representation of multiple project views as well as the interrelationships between them. The authors provided three primary dimensions (project views) to be used as the primary project coordination mechanism, and the definition of the interrelationships between them defines a three-dimensional space, as illustrated in Figure 1.1 [Froese, 2].



**Figure 1.1: Schematic of the Dimensions in a Unified Approach to PM [Froese, 2]**

By adding more interrelated project views, the unified framework can be extended to a multidimensional space, which may be represented through a set of two-dimensional inter-linked matrices. The practical requirement for applying UPM is

some kind of IT platform that supports the presentation of project information in an integrated multi-dimensional nature.

By and large, the industry need can be concluded as a need for a better analysis-driven multi-view management environment. The communication or integration of various types of information is valuable to the current project management practice, although it does require special attention in the establishment of project control procedures and in the presentation manner of project data.

### **1.2.2. Technology Opportunities**

Considering the industry needs, the IT tools that are most commonly used in current project management practice seem to be single-task and lack an integrated view of the comprehensive project status from multiple perspectives. These IT tools offer little help to shift the current improper project management thinking and practice due to the inefficient representation and analysis of project information.

The recent IT advancements in data storage, processing, and dynamic displaying have increased the opportunity to develop a fundamentally new system for project control. In fact, some of the IT advancements have already been widely utilized in other domains. For example, OLAP technology, or “On Line Analytical Processing”, has been used to control business activities for almost 20 years. Based on OLAP, “Business Intelligence” or “BI” solutions are beginning to be widely used to convert data into information that can be used to make timely, informed, and clear business decisions. In contrast with conventional data reporting, OLAP is much faster, more flexible and thorough. If properly designed, it minimizes the requirements for custom analysis programming.

There is a potential to integrate OLAP technology with management in construction, just as with the management in business. In that case, we may move towards “Project Intelligence” solutions, analogous to “Business Intelligence”.

### **1.2.3. Summary of Research Motivation**

The research motivation comes from two parts – industry needs and technology opportunities. The current project management practice in the construction industry needs to be refurbished to a more unified and multi-dimensional framework (UPM) [Froese, 2], while the development of Information Technology may provide opportunities, such as using OLAP, for a new PM tool that can make this framework applicable.

### **1.3. Research Objective**

The thesis research intends to apply the idea of OLAP-based “Business Intelligence” (BI) to project management practice in the construction industry for a potential OLAP-based “Project Intelligence” solution. The objective is to test and evaluate whether OLAP technology can facilitate project management in construction, at least in some selected aspects.

Because the multi-view UPM framework matches with OLAP’s multi-dimensional data structure, the proposed OLAP-based prototype may also be regarded as a first step for an applicable UPM framework. If OLAP does have the ability to show multiple views with their interrelationships, then OLAP may be an ideal technology for implementing the UPM framework.

### **1.4. Research Scope**

Because of time and resource limitation, as well as the extreme complexity of the system, the proposed OLAP-based PM prototype cannot be thorough and all-inclusive. The prototype is for testing and concept demonstration, rather than direct applications in the real-world industry. There are two specific PM areas that have been selected for system testing and data analysis: project cost control and project manpower allocation. There are several assumptions and design simplifications for this prototype, which are introduced

and discussed in detail in Chapter 5.

### **1.5. Prototype Features**

The proposed OLAP-based cost control and manpower allocation analysis system should be able to perform all tasks that the current practices accomplish. Besides, in contrast with the conventional solutions, the pilot OLAP-based prototype is expected to have the following main features and improvements:

- Improve the flexibility and accuracy of quantitative data (costs and man-hours) presentation in an integrated multi-view manner;
- Reduce the vagueness of implicit interdependencies between multiple project views, providing an analysis-driven environment for various aspects of the project status;
- Reduce the responding time to do project controlling analysis.

## CHAPTER 2: OVERVIEW OF OLAP TECHNOLOGY

### 2.1. Introduction

Since the mid-90's, the OLAP technology has been widely used mainly in business management and analysis. According to OLAP Report<sup>2-1</sup> the estimate of the worldwide total OLAP market, including implementation services, will reach about \$6 billion in total by 2007 (Figure 2.1).

This chapter briefly introduces the general concepts relating to OLAP and multidimensional technology. Most of the definitions, illustrations, and narratives in this chapter are from the Microsoft® website or Microsoft MSDN Library.<sup>2-2</sup>

For better understanding, the examples selected in this chapter are based on business applications. The mapping of generic OLAP concepts with construction project management applications will be discussed in the next chapter.

### 2.2. Overview of OLAP and Data Warehouse

The term OLAP stands for On-Line Analytical Processing. Developed by Dr. E.F. Codd, with colleagues Clynch Salley and Sharon Codd, OLAP was defined as “the dynamic synthesis, analysis, and consolidation of large volumes of multidimensional data”. For around 10 years, OLAP-based data processing and a new product category for “multidimensional analysis” have been greatly developed, especially in the business area.

Generally OLAP is regarded as a category of applications and technologies for collecting, managing, processing and presenting multidimensional data for analysis and management purposes [Pendse, 3]. In most instances, OLAP queries can respond much faster than

---

<sup>2-1</sup> The OLAP Report is an independent research resource for organizations buying and implementing OLAP applications. <http://www.olapreport.com/>

<sup>2-2</sup> The Microsoft® Website: <http://www.microsoft.com>. The MSDN Website: <http://msdn.microsoft.com>.

relational ones, even for large-sized databases with high complexity. More importantly, OLAP queries seem to be much more flexible, which allows any dimension, level or member to be accessed and returned to either a row or column in a result set, providing superior performance for ad hoc business intelligence queries.

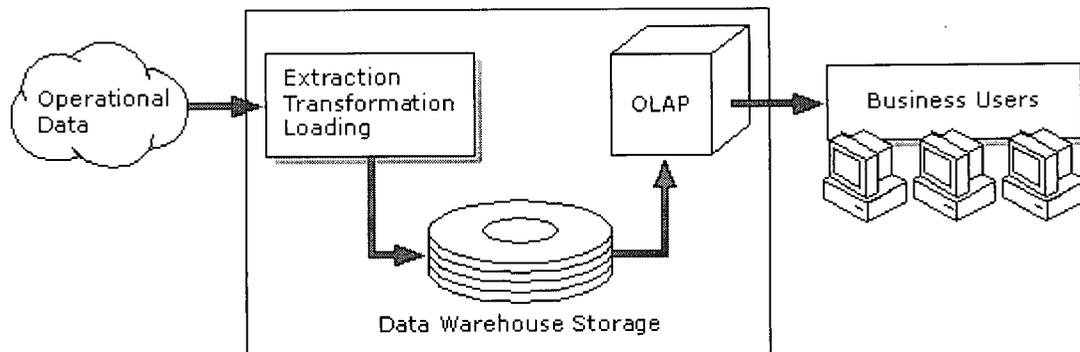


**Figure 2.1: Worldwide Total OLAP Market**

OLAP technology provides remarkable performance in rapidly summarizing information for analytical queries, dealing with data organized in accordance with the common dimensional model maintained in a data warehouse. A data warehouse is a database that usually contains stable and consistent business history of an organization. According to MSDN Library, “The purpose of a data warehouse is to consolidate and organize historical data so it can be dynamically captured and analyzed to support business decisions” [Microsoft, 4]. OLAP technology enables data stored in a data warehouse to be used effectively for online analysis, providing rapid responses to iterative complex

analytical queries. OLAP adopts multidimensional data models and data aggregation techniques to organize and summarize large amounts of data so that they can be evaluated quickly using online analysis and graphical tools.

Figure 2.2 [Microsoft, 5] shows the basic structure of a typical OLAP-based business intelligence system.



**Figure 2.2: Business Intelligence (BI) with OLAP & Data Warehouse [Microsoft, 5]**

Generally operational data is stored in Online Transaction Processing (OLTP) databases that are designed for real-time operations, and optimized for a common set of transactions, usually adding or retrieving a single row at a time per table. Such operational databases are designed to gather and store data processed or generated by all business operations. A data warehouse offloads the historical data through extraction, transformation, and loading processes (ETL). In this way, high volume analytical and reporting queries are handled by the data warehouse with no impact to the original operational databases. The inherent stability and consistency of historical data stored in a data warehouse ensures accurate and rapid responses for analytical queries.

The OLAP technology organizes data warehouse data into multidimensional data cubes and then pre-processes these cubes to provide maximum performance for queries that summarize and present data in various ways. It is commonly implemented in a multi-user client/server mode and offers rapid response to queries regardless of the size and complexity of the database. The whole system enables businesses to merge raw

information together for comparative and personalized viewing, as well as for analyzing historical and projected data.

## **2.3. Key Concepts in OLAP and Data Warehouse**

To look further into OLAP technology, it is critical to understand several key concepts within this area, including cubes, dimensions, levels, members, measures, MDX, and more.

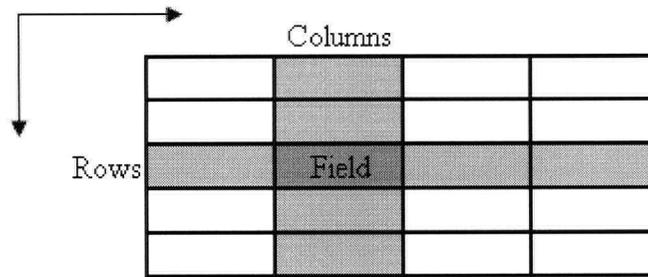
### **2.3.1. Dimensions, Levels, and Members**

According to the definition from Microsoft MSDN library, a dimension is “an organized hierarchy of categories (levels) that describes data in data warehouse fact tables” [Microsoft, 6].<sup>2-3</sup> It is a fundamental component of cubes that typically describes “a similar set of members upon which the user wants to base his or her analysis”.

From this perspective, traditional relational database, along with most languages used for data definition and manipulation such as SQL, is designed to access data in two dimensions: a row dimension and a column dimension. At the intersection of each row and column is a single element of data, called a field (Figure 2.3). Either the row dimension or the column dimension is actually one category for the data, thus one dimension can be regarded as one perspective that is used for looking at the data. In a sense, dimensions are the answer to a question such as “How do you want to see your data?”

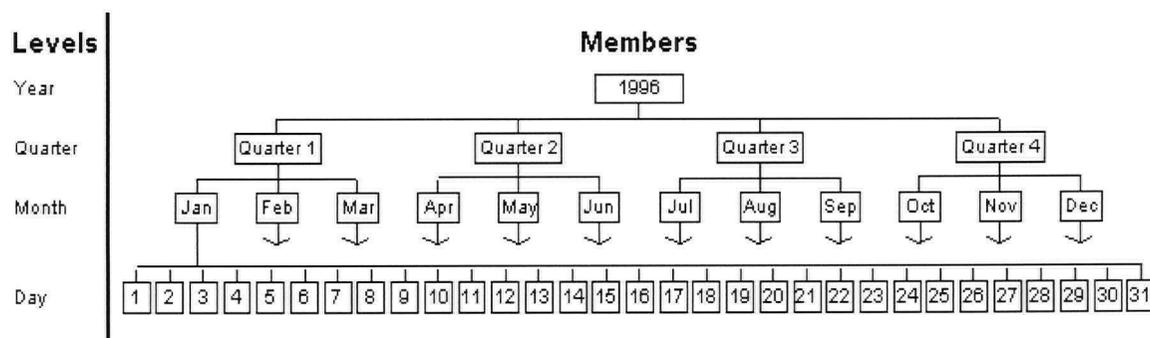
---

<sup>2-3</sup> A basic schema for data warehouse is several dimension tables plus one fact table. Detailed introduction of multidimensional database structure refer to section 2.4.1.



**Figure 2.3: Traditional two-dimensional data structure**

For a traditional two-dimensional spreadsheet, both the row dimension and the column dimension consist of members that are organized in one level. Those members are generally presented as row/column labels in a spreadsheet. For multidimensional data structures in OLAP, however, dimensions are “hierarchical and in most cases their members are arranged in different levels of a pyramid-like configuration”. “The horizontal placement results from column values with the same level in the hierarchy of the dimension, and the vertical placement results from column values having different levels in the hierarchy of the dimension” [Microsoft, 7]. According to Microsoft MSDN, “level” is the name of “a set of members in a dimension hierarchy such that all members of the set are same distance from the root of the hierarchy” [Microsoft, 8]. For example, a time dimension (Figure 2.4) may contain 4 levels including Year, Quarter, Month and Day [Microsoft, 9].



**Figure 2.4: A Typical “Time” Dimension [Microsoft, 9]**

With levels, OLAP describes dimension hierarchy from the highest (most summarized)

level to the lowest (most detailed) level of data, making it possible for users to see a general view of things from a high level, as well as a detailed view of things from lower level. The users can select data they find interesting at a high level and drill through to the source data to view extra detail. They can also remove the data with one particular member from one particular level, or combine different dimensions together to present more interesting data.<sup>2-4</sup>

### **2.3.2. Measures and Aggregation**

Compared with dimensions, which are views to organize data, measures are the actual data values that are stored in a multidimensional data cube—as if the data value stored in the “field” areas of a two-dimensional spreadsheet. They are the central values that are aggregated and analyzed. For example, a retail cube may contain sales values that allow you to view sales (measure) according to time (dimension), region (dimension), and product (dimension).

Measures are generally numeric. By themselves, pure numbers do not convey meaning. For instance, the number 1200 stored in a cube does not have meaning by itself. However, when relevant members from different dimensions have been identified, the number will take on more meaning. For the above retail cube for example, 1200 stored in this cube may represent the total sales of snowshoes (product) in Toronto (region) during December 2004 (time).

OLAP technology provides rapid query responding times, and the basic foundation of this feature is aggregation. Aggregations are pre-calculated summaries of multidimensional data that greatly reduce query response time by having answers ready before questions are asked [Microsoft, 10]. Again, take the above retail cube example. When creating the cube, we take every detailed sales data and add them up in advance by city, by province,

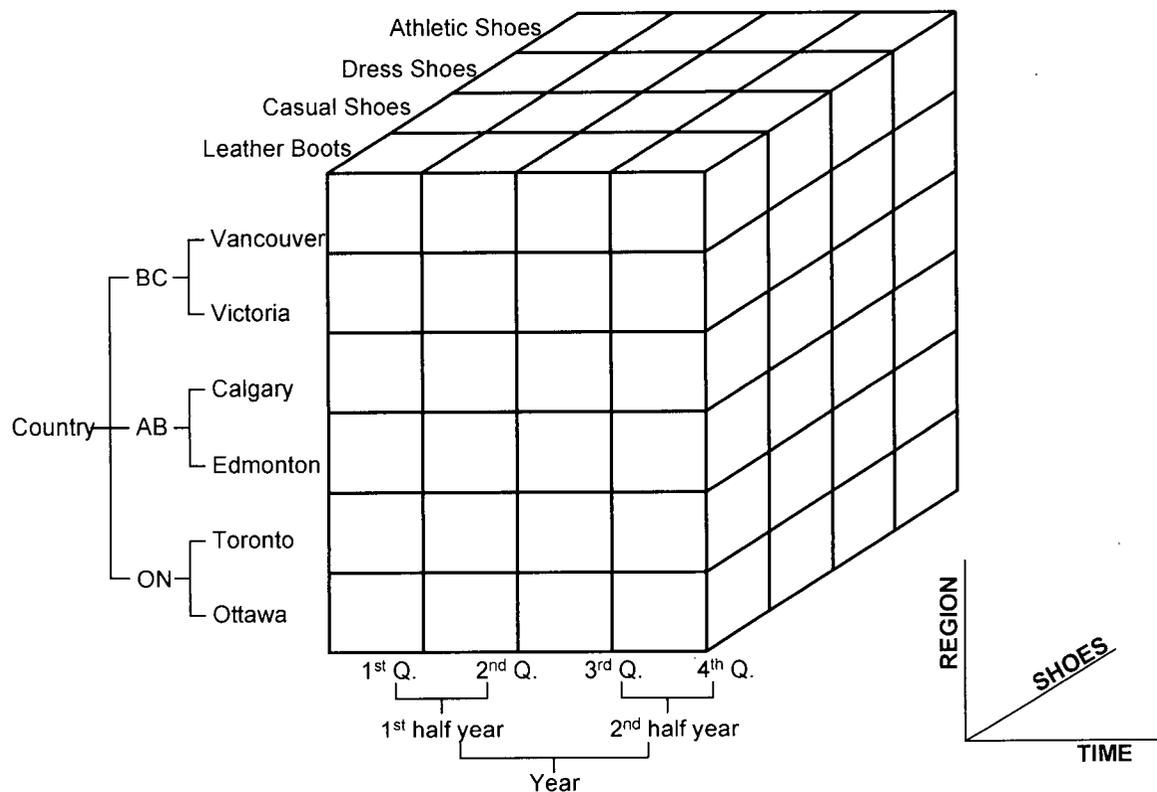
---

<sup>2-4</sup> These OLAP operations are known as “Drilling” and “Slice & Dice”, which are further discussed in section 2.4.

by month, by year, and by any possible combination of dimensions and hierarchy levels. Now if the user asks for total sales by city and month, or by province and year, the numbers are already available. Because of this, the response of an analytical query can be returned in no time since the summarized data to answer the query has already been pre-calculated.

### 2.3.3. Data Cubes

Basing the above key concepts of dimensions, levels, members, measures, and aggregations, the concept of cubes is easy to understand. A Cube is the method OLAP technology organizes pre-processed summary data into multidimensional structures [Microsoft, 10]. In other words, cubes provide a multidimensional way to access and analyze data. A cube aggregates the facts in each level of each dimension in a given OLAP schema, and aggregations are stored in the multidimensional structure specified by the dimensions [Microsoft, 10].



**Figure 2.5: Concept Model of the “Retail” Data Cube**

Figure 2.5 shows a typical conceptual model for the foregoing retail cube and presents the multidimensional view of data with 3 dimensions—time, region, and shoe product. The “Shoe Product” dimension has only one level while both the other two dimensions have 3 levels. The term “cube” here seems to imply a three dimensional structure, but actually in the OLAP world a cube can have up to 128 dimensions, each of which may have thousands or millions of members. Those dimensions are either private to the cube itself or shared with other cubes in the database. Different to dimensions, the measures in a cube can only be private and are not shared with other cubes. One cube can have up to 1,024 measures at the most.

Just as data can be accessed from a two-dimensional spreadsheet at the intersection of one particular row versus one specific column, the preceding cube diagram illustrates that the intersection of multidimensional members creates a cell or several cells where data can be retrieved. Regardless of it being a single cell or a block of cells, MDX (Multidimensional Expression) uses a reference system called “tuples”, through listing all dimensions and members, to identify and extract such data. Tuples can uniquely identify every cell because each of the cells is an intersection of all dimensions of the cube [Microsoft, 11].

The schema of data cubes will be further discussed in section 2.4 along with the issue of multidimensional data analysis.

#### **2.3.4. MDX & PivotTable Services**

“Most languages used for data definition and manipulation, such as SQL, are designed to retrieve data in two dimensions, i.e. a row dimension and a column dimension” [Microsoft, 11].

The Multidimensional Expression (MDX) language, on the other hand, was created with the primary purpose of allowing the user to manipulate multidimensional data, which are

organized by structures with more than two dimensions. It's used with Microsoft SQL Server Analysis Services as a query language, and is used by various client components and applications to return multidimensional data from OLAP cubes. The language is called an "expression" because it can be used in SQL Analysis Services to define calculated members, i.e. members that are not stored in the cube but can be derived from the existing members and measures. In Microsoft SQL Server 2000 Analysis Services, MDX can also be used to build local cubes and to query cube data using the PivotTable Service [Pearson, 12].

According to the Microsoft MSDN Library, PivotTable Service is an OLE DB provider for multidimensional data and data queries. "This means that it provides OLE DB functionality for applications that need access to multidimensional data and data mining services. By providing support for a subset of SQL and Multidimensional Expressions (MDX), PivotTable Service enables applications to retrieve tabular and multidimensional data" [Microsoft, 13].

PivotTable Service acts as a bridge between a client application and a multidimensional data source. This thesis research will use Office Web Component (OWC) as the client application for data presentation and analysis, which is further introduced in section 2.5.

#### **2.4. OLAP Database and Multidimensional Data Analysis**

A data cube is the main object in OLAP technology that provides an easy-to-use mechanism for querying data at high speed. As stated, OLAP provides a remarkable performance on query response time because the cubes pre-process all data in an aggregated form. That way all answers will be created before potential questions are asked.

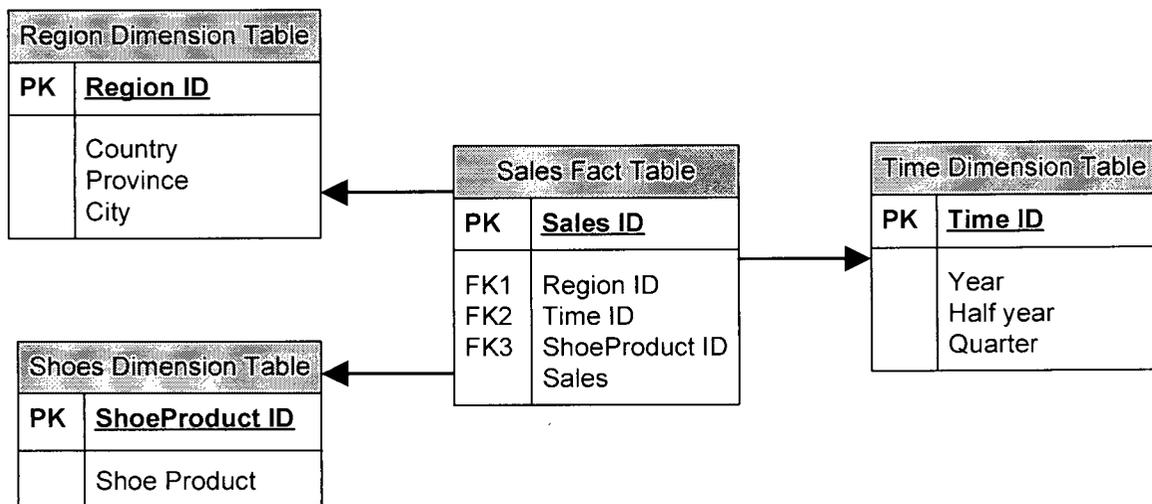
This section is to introduce the basic cube schema and show how pre-calculated summary data (aggregations) are created. Also included in this section is cube data analysis, which

focuses on the OLAP operations, known as “Slice & Dice”, “Drilling”, and “Pivoting”.

### 2.4.1. Cube Structure and OLAP Database

A cube's structure is defined by its measures and dimensions mentioned above. They are basically derived from  $n+1$  tables in the cube's data source, which include a single “fact table”, and  $n$  “dimension tables”. The cube's measures are all derived from columns in the fact table, and the cube's dimensions are all derived from columns in the dimension tables.

For the retail cube example, the cube may have the schema shown in Figure 2.6.



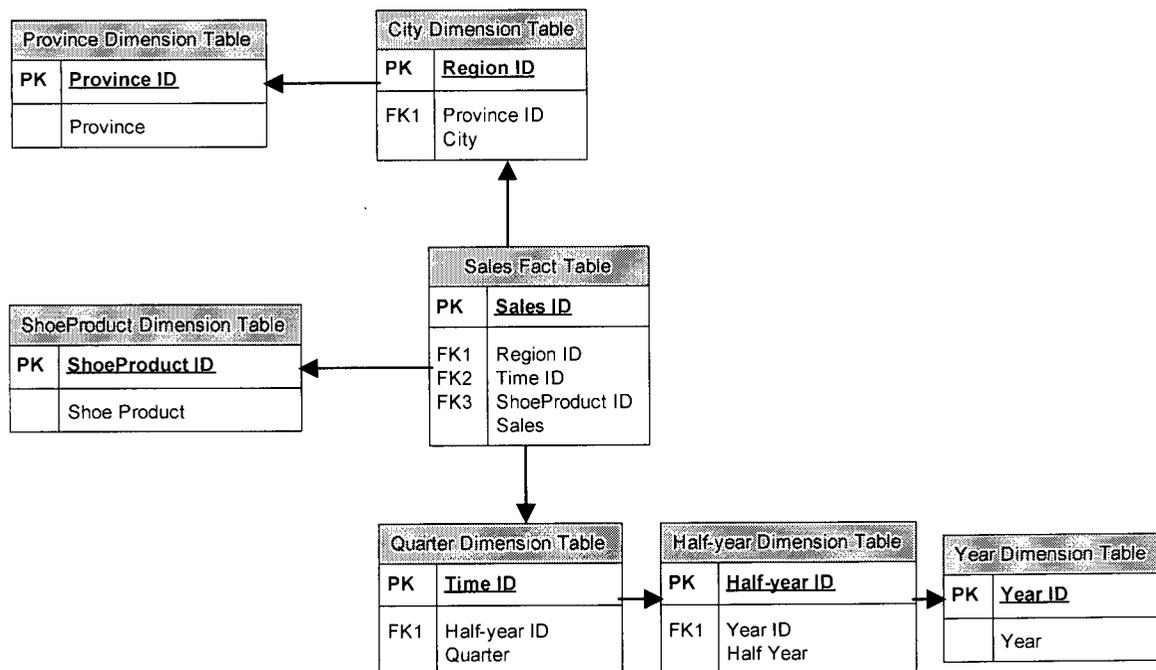
**Figure 2.6: A “Star” Schema of the “Retail” Cube**

Figure 2.6 shows a star schema that is generally used to represent the multidimensional data model. The fact table usually sits in the middle of the star. Each tuple in the fact table consists of a pointer (foreign key) to each of the dimensions that provide its multidimensional coordination. The combination of those referred dimensions can uniquely determine the measure, or measures. For example, the following Table 2.1 shows a single cube cell that is derived from multiple rows in the fact table. The cell is in the above “Retail” cube for the member of “Ottawa” in the region dimension, the member of “1<sup>st</sup> quarter” in the time dimension, and the member of “Dress Shoes” in the shoes product dimension are derived from the Sales fact table. The same values of the Region ID, Time ID and ShoeProduct ID indicate that these rows contribute to the same cube cell.

**Table 2.1: A sample cube cell from the “Retail” cube**

Sales ID	Region ID	Time ID	ShoeProduct ID	Sales
17284	6	1	3	\$42,431
23843	6	1	3	\$23,123
42432	6	1	3	\$20,392

A classic star schema meets the condition that all dimension tables can be joined directly to the fact table. In case “one or more dimension tables do not join directly to the fact table but must join through other dimension tables”, the schema is called a snowflake schema [Browning, 14]. Snowflake schemas provide a refinement of star schemas, and the dimensional hierarchy is explicitly represented by normalizing the dimension tables, as shown in Figure 2.7. By breaking hierarchical levels into separate tables, referential consistency between the levels of the hierarchy is guaranteed. This results in advantages in maintaining the dimension tables, especially in case of a very complex dimension [Browning, 14].



**Figure 2.7: A “Snowflake” Schema of the “Retail” Cube**

### **2.4.2. Slice & Dice**

Even for multidimensional data models, the spreadsheet is still the most compelling client application for OLAP. “Slice and Dice” is one of the most popular operations that are supported by the multidimensional spreadsheet applications. “Slice and Dice” corresponds to reducing the dimensionality of the data, i.e. taking a projection of the data on a subset of dimensions for selected values of the other dimensions [Chaudhuri, 15].

To speak specifically of spreadsheet applications, the users can use one dimension for slicing when it isn't being displayed on the columns or rows of the spreadsheet, and they can slice on any member from any level of the dimension. For example, if the user slices on member “Vancouver” in the city level of the “Region” dimension, he or she can focus the view of data only on those relating to Vancouver. On the other hand, the user can cut with sets of members from two or more dimensions, which is called “Dicing”, by putting more than one dimension on the row or on the column of the spreadsheet in a single scenario. That way the user will be able to see from the spreadsheet every possible combination of all members from each of the selected dimensions.

### **2.4.3. Drill up & Drill down**

Levels of each dimension provide the user opportunities to review data at a higher level as well as opportunities to explore into the details of the data. Drill-up & Drill-down refers to the change of aggregation levels along one or more dimension hierarchies. Drill-up (or Rollup) corresponds to taking the current data object and doing a further group-by on one of the dimensions, and Drill-down is just the converse of Drill-up.

Using Drill-up and Drill-down for the retail cube, it is possible to roll-up the sales data, which perhaps has already been aggregated by city, by time period, or by product. After viewing the sales of 2003 in a particular province, for example, the user may want to drill down to see situations in a particular city. If the user notices that the sale value of casual shoes in Vancouver at the 1st quarter of 2003 is extraordinarily low, he or she may then

want to drill up to see if the data patterns are still valid on a higher level.

## **2.5. Office Web Components (OWC)**

There are various client components and applications used to return multidimensional data from OLAP cubes, among which Office Web Components (OWC) are a set of powerful and easy-to-use tools for OLAP data representation and analysis. OWC is a set of components provided by Microsoft that can help build sophisticated but user-friendly OLAP-based reporting solutions for the Web. It is a collection of Microsoft ActiveX controls that can be embedded in intranet, Internet, or desktop applications to provide Office-like functionality at the user level. Through OWC, users can add, change, sort, or filter data in their own way. They can also expand and collapse detail levels, deal with PivotTable lists, and develop charts in various formats in accordance with the data. For developers, OWC offers a way to improve user productivity by making it easy to create Web applications that solve a variety of problems.

There are four types of OWC components: a PivotTable Component, a Spreadsheet Component, a Chart Component and a Data Source Component. Table 2.2 [Microsoft, 16] shows a summary description of all the Office Web Components (Version 10) that ship with Microsoft Office XP.

As for a proper OLAP-based front-end client application, an ideal choice is the PivotTable component that brings the ability to “slice and dice” data within a Web browser. The PivotTable Component also allows users to analyze data by adding the sorting, grouping, filtering, and other possible data manipulation functionalities to Web Pages. It is roughly equivalent to a PivotTable report in an Excel worksheet, except it is mainly used for web-based OLAP solutions.

**Table 2.2: Summary of OWC Components [Microsoft, 16]**

<b>Components</b>	<b>Description</b>
PivotTable	This component makes it possible for users to sort, group, filter, outline, and manipulate data from a worksheet, database, or multidimensional data cube.
Spreadsheet	This component provides a recalculation engine, a full function library, and a simple worksheet user interface for use on Web pages.
Chart	This component displays a graphical representation of data from a Spreadsheet, PivotTable List, or Data Source control. When bound to other controls on a page, the Chart control updates instantly in response to changes made to the bound controls.
Data Source	This control manages communication between a Web page or controls on the page and the source of data for the page. This control provides the reporting engine behind data access pages as well as the PivotTable List control.

In summary, the benefits of using the Office Web Components include the following:

- 1) They support web-based deployment. Office Web Components deliver advanced analysis capabilities in intranet/internet solutions, making OLAP solutions accessible to multiple users from the widest possible user base.
- 2) They access OLAP-based data. Like Microsoft Excel, Office Web Components support OLAP cube data analysis. Developers can build dynamic analysis solutions that connect directly to OLAP data sources.
- 3) They lead to extensible and fast development. Office Web Components enable developers to make the most of their familiarity of Microsoft Office, putting the power of windows-based Microsoft Excel in their web-based solutions. For developers that are familiar with Office development environment, Office Web Components make the development process quick and effective. With the help of

those pre-packaged analysis components, developers can focus on assembling an analysis solution instead of coding and implementing the operations.

- 4) They ensure high performance on the client side. Office Web Components ensure fast and efficient performance over an internet connection for clients because the analysis logic runs on the server side. End users are not required to maintain a connection to the Microsoft SQL Server database. As long as Internet Explorer and OWC are installed on the client computers, the users can view OLAP data stored on database server. (Note: OWC version 10 is shipped with Microsoft Office XP. Users do not need to install OWC if they have Office XP on their computers.)

## CHAPTER 3: PROJECT MANAGEMENT & OLAP TECHNOLOGY

### 3.1. Introduction

The objective for this thesis research is to evaluate the possibility of taking advantage of OLAP technology to improve project information analysis, just as OLAP-based BI solutions do in business management. This chapter serves as the first step to map the generic OLAP concepts stated in Chapter 2 with general project management concepts in the construction industry.

Also, the Unified Project Management (UPM) approach, as stated in Chapter 1, looks at the total collection of project information as a multi-dimensional information space. OLAP technology transforms relational data into a multi-dimensional data structure (Cubes) for flexible data presentation, fast query response, and intelligent data analysis. So, if it is possible to store project data in a data warehouse as a multi-dimensional data structure, then an OLAP-based PM platform may be a good way to support the proposed UPM framework.

### 3.2. Related Research

#### 3.2.1. Research on Integrating Construction Information

Construction projects do need an effective communication infrastructure to facilitate information interchange between all the project participants, and so far, several research projects have been focused on this area.

- *ICON*: The ICON (Information / Integration for Construction) Project [Aouad et al., 17] was developed at the University of Salford from UK, leading to the establishment of a framework for integrating information systems in the construction industry. The project focused on the development of a set of models that were integrated within a central object-oriented database, which could be

used as an interface between the different disciplines to share information [Aouad et al., 17].

- *COMMIT*: The COMMIT (Construction Modelling and Methodologies for Intelligent Information Integration) project was also developed at the University of Salford on top of ICON project. The overall aim of the COMMIT project is “to improve the long-term effectiveness of the construction industry by the provision of intelligent integration of information to support decision making for the effective management and realisation of all stages of design and construction” [Rezgui, 18]. The COMMIT mainly focuses on distributed information management between different project participants, through a COMMIT Information Management Model (CIMM) based on a distributed database.
- *WISPER*: The WISPER (Web-based IFC Shared Project EnviRonment) is a Web and Industry Foundation Classes-based (IFC-based) distributed computer integrated environment, adopting a three-tier client–server infrastructure to share information through an IFC object-oriented project database. WISPER is applicable to detailed design and construction phases, and it incorporates a set of Web pages allowing for remote interaction and access to distributed applications [Faraj, 19].

### **3.2.2. Research on Multidimensional Data Modelling for Construction Information**

Each of the construction information integration systems introduced above is based on a central or distributed object-oriented database. Although data warehouse development, the OLAP technology, and the multidimensional data modelling and presentation, have been widely used in helping business managers or decision-makers analyze enterprise data, known as Business Intelligence (BI), the effort on multidimensional data modelling for construction information integration is still comparatively limited.

- A Construction Management Decision Support System (CMDSS) was introduced in 2002 [Chau, 20], as a data warehouse integrated information management prototype. The CMDSS was designed to separate the analysis database from the operational database, and use conceptual multidimensional data model for dynamic data presentation. The authors introduced the basic concepts and the design process for a data warehouse to support the system. The benefits of adopting data warehouse and the OLAP technology, as well as the considerable potential of further development, were pointed out. However, the authors did not provide a general mapping of the OLAP technology to project management in the construction industry. There was a lacking of analysis on what information categories in construction industry can technically take advantages of the OLAP technology through multi-dimensional data presentation, and how the improved data representation can help information analysis and decision making.
- In a journal paper [Songer, 21] published in April 2004, the authors pointed out that the construction industry lacks a fundamental model of project data representation, and the main reason is “the inadequate application and implementation of visual tools in project control methods”. The authors provided a framework for developing visualization strategies for multi-dimensional data control, and illustrated the development of cost/budget data visualization layouts using a treemap strategy. The authors did not adopt the OLAP technology and the data warehouse concepts in the paper, but their discussion regarding the data structure of a multi-dimensional view of project data was thoughtful and convictive.

### **3.3. Overview of Mapping OLAP to Project Management**

This thesis research is trying to use the idea of Business Intelligence for a “Project Intelligence” approach. As opposed to accessing and presenting business data for analysis,

Project Intelligence derives data from project-related information sources, assisting project personnel to understand the performance of projects, and conduct project information analysis.

The main differences between the “project” data and the “business” data are in the variety and the complexity. The business data generally involves a high volume of data. But in terms of the data type and the interrelationships among data, the business data is comparatively simple and straightforward, compared with the project data. Regardless of the business type, the branch locations, or the product category, the fundamental business data (defined as the “measures” in an OLAP data cube) is almost always the cost such as—the budgets, the revenue, or the forecasts. The external information linked with the central data (defined as the “dimensions” in an OLAP data cube) is also easy to define and express—such as the “store location”, the “product type” or the “time”.

The project control data, however, is much more complicated. Generally the amount of information produced in construction processes is huge. Some information is hard to define and express, such as the information relating to risk, quality, or product. On top of that, all project data are interconnected with vague relationships. All of these make the definition of OLAP “dimensions” difficult. In addition, because not all information produced in construction is numeric with a consistent unit, it is sometimes very hard to define the OLAP “measures” too. *(A detailed mapping of project data to the OLAP concepts is further discussed later in the chapter.)*

When properly designed and integrated, however, the OLAP technology may be able to provide project participants with the ability to visualize project data through various project-related dimensions in a familiar, easy-to-use environment such as Microsoft Excel. Using PivotTable services provided by Microsoft Office Web Components, project team members can get web-based access to dynamically view, organize, and run queries on project information from anywhere with an internet connection.

After organizing project data into OLAP cubes, project team members can access project information from any member of any dimension through “slice and dice”, or they can view project information on various levels through “drill-up and drill-down”. For example, noticing the labour allocation is extremely high on one particular day, the project manager may drill down the participant dimension to see which one contributes the most among all organizations working that day. After figuring out that the concrete crew has most labours on-site, he or she may then want to slice on the concrete crew to view its overall labour allocation by day, or by task on a wider scale.

Even for a small-sized project there may be various kinds of project information. Not all of them, however, are suitable to be organized in OLAP cubes as dimensions or measures. The reason may be technical, i.e. OLAP technology does not support the data type. It may also be non-technical, i.e. the data can be plugged into an OLAP cube and run properly, but the presentation of data does not make sense or does not help much for analysis. It is the system designer’s job to choose appropriate project data as dimensions and measures. He or she may also have to consult with the final users (project team members) before cube design, ensuring that the final presentation of data makes sense and is helpful for analysis.

### **3.4. Dimensions**

General speaking, a dimension in OLAP is a particular way or view that can be used to observe data. It is a set of attributes that roughly correspond to RDBMS fields. To some extent, dimensions are the answer to a question such as “How do you want to look at your data?”

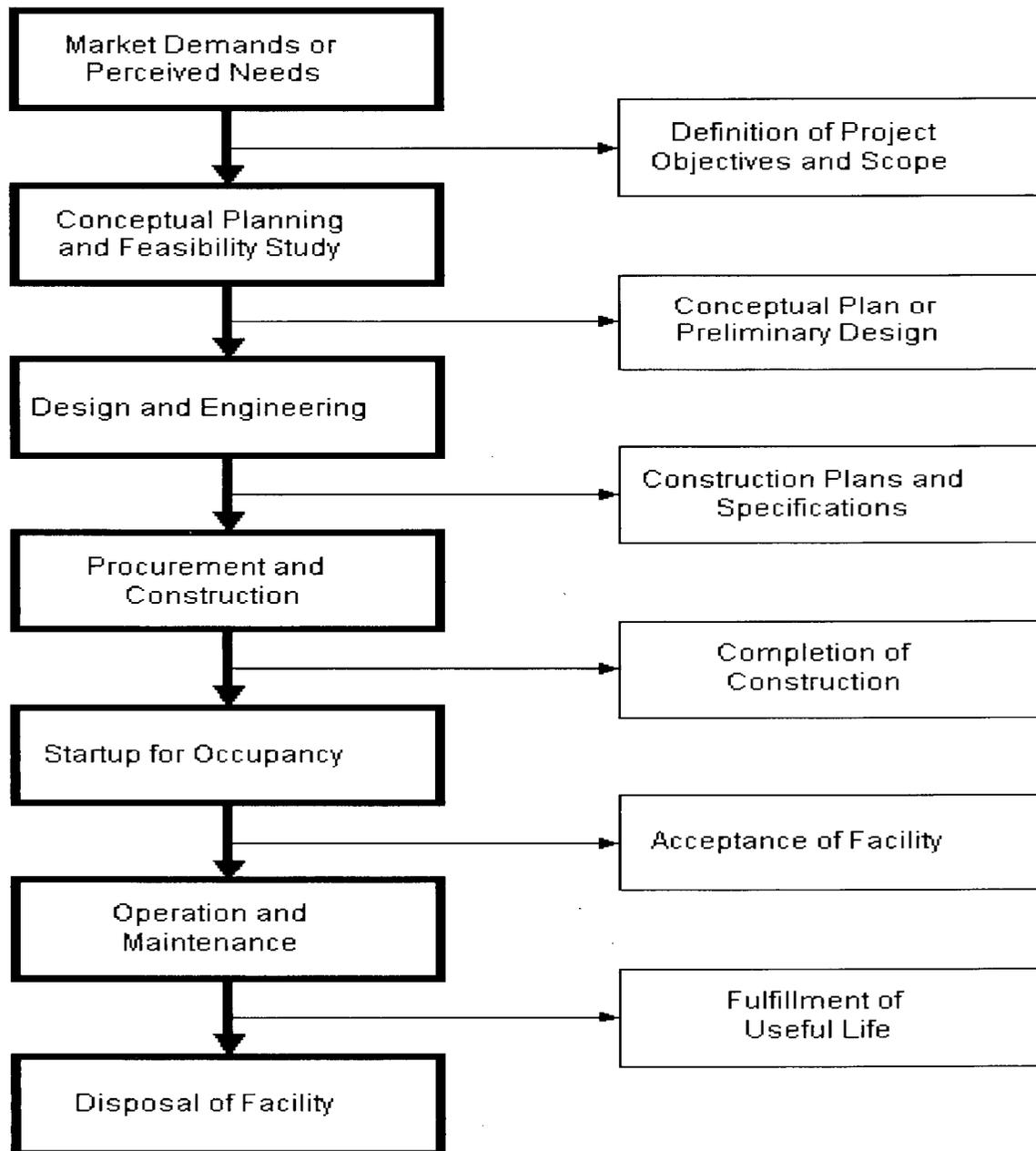
For project management purposes, there may be many distinct ways that we would like to look at the project data. Different people may have different foci, but there are still some common views that interest all participants. Those views will be defined as dimensions

when developing the multi-dimensional project data structure. Following are some examples for project management dimensions:

#### **3.4.1. Time Dimension**

One of the critical objectives for project management processes is to deliver the project on time, so the time dimension will be of interest to almost all project participants.

A construction project can be organized into well-defined phases during the project life cycle. The project life cycle may be viewed as a process through which a project is implemented with a breakdown of several stages as indicated by the general outline in the following Figure 3.1 [Hendrickson, 1].



**Figure 3.1: The Project Life Cycle of a Constructed Facility [Hendrickson, 1]**

Each activity in an AEC/FM project occurs at a specific date and within a specific period. The time data stored in a data warehouse is often summarized by a specific time period for analysis, and the Time dimension provides an effective mechanism for time-oriented analysis.

The breakdown of project life cycle into different phases (time periods) as shown in Figure 3.1 may be a little too fuzzy to be used for data analysis because each stage may have a long time span and contain too much information. For management purposes, we may further decompose each stage—but a Time dimension with one record per day would be sufficient. The smallest time unit used in the construction industry for scheduling, resource allocation, procurement arrangement, etc. is “day”. In construction management processes we seldom need time granularity finer than a single day. If this is the case, a dimension table for the Time dimension, or say the “Date” dimension, will contain 365 records per year.

Of course, the development of the time dimension depends on how the potential project data analyst—whether the owner, the project manager, or other team members—would like to format the data presentation. If a well-defined hierarchy for time dimension is necessary, we can define levels and organize the time-based project data by week, by semi-month, by month, by quarter, by half-year, by year, or by any other time intervals that seem reasonable. In most cases, in the construction industry, a “day-month-year” hierarchy for the time dimension can meet analysis requirements.

### **3.4.2. Activity Dimension**

As opposed to the time dimension, the activity dimension is a process-based view of project management. It organizes the whole project into various work disciplines required from the beginning of the project to completion.

If we observe project processes from this perspective, we actually split the whole project into different work divisions such as bidding work, design work, procurement work, construction work, and so on. Each work division may be involved with many organizations or participants, and each division can be further split into several subdivisions. For example, the construction work division may have subdivisions such as excavation, foundation work, formwork, placing rebar, pouring concrete, etc. These

subdivisions may be further broken into even smaller tasks or subtasks.

Compared with the time dimension, the definition of the activity dimension seems to be more difficult, especially in terms of defining levels and members using OLAP. The reason is that for the time dimension, there are explicit definitions of the members and levels existing already from common sense. Any choice of units or organization of levels will be easily understood. Everybody knows what a “day” refers to and everybody knows one week contains 7 days. You can aggregate the summary data by week, by month, or by quarter. The only difference is the presentation of the data due to different members and levels that may have been chosen. For the activity dimension, however, the definition of the levels and members is fuzzy and largely depends on the designer’s view and understanding of the project. For instance, different people may have various understanding for the definition “concrete work”. For one person, it may mean placing concrete, but for another it may also include placing formwork and rebar. It may also include curing and formwork removal.

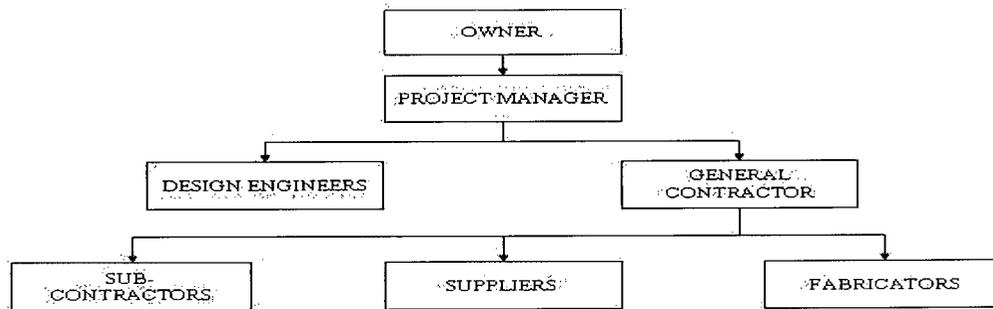
Therefore, if the OLAP cube designer will not be the final project data analyzer, it is imperative that the designer start with understanding the users and their needs before embarking upon the design.

### **3.4.3. Participant Dimension**

One critical reason why AEC/FM projects are complex is that there are always many organizations or participants involved, regardless of the type and size of project. The Participant dimension defines the project actors, and allows the project data to be viewed from the perspectives of various organizations. Because it is relatively straightforward to identify and group project participants, they provide an ideal cube dimension for OLAP.

At a high level, there are generally three participants that contribute their skills and services to form a project team – the owner, the designer, and the builder (or only two

participants if we consider the concept of a design-build contract: the owner and the design-build contractor). However, each participant group can be further divided into sub-organizations. For example the builder can be further defined as the general contractor, the subcontractors, the suppliers, and the fabricators. Figure 3.2 is the traditional organization chart for an AEC/FM project.



**Figure 3.2: Traditional Construction Participants**

Figure 3.2 only shows the relationships between each participant. It does not show the level definition. When designing the Participant dimension for an OLAP cube, the general contractor stands at the same level as the subcontractors, the suppliers and the fabricators. In other words, there is no parent-child relationship between them. Only the sub-organizations within a particular organization can claim such kind of relationship. For example, only the sub-organization such as the concrete crew, or the rebar crew, stands at the lower level beneath their parent organization—the general contractor. The subcontractor stands at the same level as the general contractor, but it may be further recognized in lower levels as an excavation subcontractor, a ceiling subcontractor, or a windows subcontractor.

#### **3.4.4. Product Dimension**

To some extent, construction management is largely product oriented. The primary goal for a successful project management process is to deliver the expected construction product on time, within budget, and with satisfactory quality. The product dimension

organizes project data into various outputs or deliverables that make up the final project system, whether a building, dam, bridge, sewage treatment plant, or any other type of projects/facilities.

Product here can be virtual or physical. In the early phases of a project, the information packages that describe the construction product being built can be regarded as virtual products. These may include contracts, design drawings, purchase orders, and other possible information packages. After contractor mobilization and construction start up, products become the physical elements themselves like columns, slabs, or stairs.

As with the activity dimension discussed above, the definition of production dimension in OLAP is also discretionary and greatly depends on the project type and the designer's viewpoint. For a high-rise condominium building project, the "construction" member of the product dimension defined in OLAP may be further divided by stories from the foundation to the roof. For a mining plant, on the other hand, the same member of the OLAP product dimension may be organized by areas such as crushing area, grinding area, storage area, loading area and so on. Even for a single project, different designers may have distinct ways to define the levels and members in this dimension.

### **3.5. Measures and Aggregations**

Measures are the actual data values that are stored in the data warehouse or a data cube. They are the data which project managers, project participants, or data analysts may be interested in, and thus they are the data that need to be clearly presented in a dynamic and intuitive manner for better control and analysis. If we use an OLAP-based spreadsheet (PivotTable for instance) as the front-end application, measures are the numbers that are displayed in the cells of the spreadsheet.

As stated in chapter 2, the information used as measures is contained in a relational database table which is called the fact table. Whenever we create a new cube, the first

step is to identify the fact table. There are a great many facts that deserve our attention, and again different people may focus their interest on different project information. However, since we are trying to improve project data analysis by adopting OLAP technology, we have to base the selection of measures on some exceptional requirements of OLAP.

There are several factors to consider when determining what kinds of project information are suitable to treat as OLAP measures and are valuable for future analysis. The following are some of these considerations.

- The data should be numeric. Measures are the central values of a cube that are the primary interests to end users browsing the cube, and they are generally numeric. Only numeric measures can be summarized by OLAP tools such as Microsoft SQL Server Analysis Services in the course of cube processing. The pre-calculated summaries of data, which is known as aggregations, can greatly improve query response time by having the answers ready before the questions are asked. The dimensions and their hierarchical levels are designed to reflect the queries that can be asked of the cube. The designer must consider what kind of numeric data that can attract the attention of project participants, and be potential interesting topics for analysis.
- The data should have the same unit. This consideration is especially important for mapping OLAP technology with project management concepts. If this consideration is lacking during cube design for project data analysis, it may lead to a data cube that produces meaningless responses to queries. For example, suppose a designer tries to develop a cube for the end users to analyze the quantity of materials consumed in a construction project. One of the dimensions for this cube is called "Material" which consists of members including "Concrete", "Rebar", and "Formwork". The measure stored in the central fact table of this

cube is the quantity for each type of material. The following data is one row sorted from the data cube indicating material consumption for one particular day.

Concrete	Rebar	Formwork
120	14	205

When processing the cube after design, a summary of 339 will be pre-calculated as the total sum of the three numbers. When the end users browse this cube from any front-end OLAP client application, the number 339 will be displayed indicating the total material consumption of that day. Obviously this will be very confusing because the three numbers that add up to the total have different units. The actual situation is that on that particular day the total consumption of concrete, rebar, and formwork is 120m<sup>3</sup>, 14MT, and 205m<sup>2</sup> respectively. Although there is no technical problem and the OLAP solution runs correctly, the summary number does not make sense.

- There should be a large quantity of data. This is certainly not a mandatory requirement. Even a situation with only two numbers in a data cube is allowed. But when selecting project data as OLAP cube measures we should make the most of OLAP's advantages, and consider whether the data is really worth future analysis. The primary advantage of OLAP technology is its remarkable performance in combining, processing, and summarizing large amounts of data in order to provide answers to questions about that data in the shortest amount of time. Instead of aggregating the data at the time the user submits a query, an OLAP cube develops pre-aggregated summary data in advance. If the data amount is very small, then the benefit of adopting an OLAP-based multidimensional model against other data structures is limited.

Based on the foregoing considerations, the possible project information that is suitable as the measure in an OLAP cube may include:

- 1) Construction Cost. The best integration of OLAP technology with construction project management is in the area of construction cost control. On one hand, construction cost information is critical and cost control is one of the paramount tasks of project management. On the other hand, cost data occurs from the start-up of construction to the completion throughout the whole project execution. It is numeric with a large amount of data—as long as we break down the whole project into pieces by each discipline such as time, task, organization, area and so on. Cost data for a construction project usually has the same units, whether it is in Canadian dollars, US dollars or any other currencies. Even if more than one currency is used for one specific project, for example a project in Canada with Canadian labours but some materials supplied from the United States, we can easily change one currency to another by multiplying the relevant exchange rate.
- 2) Construction Resource. Another interesting usage of OLAP technology in project management may be resource allocation analysis. Successful project management in construction must actively pursue the efficient utilization of labor, material and equipment. Among the three main types of construction resource, labor (Man-hours) is the most desirable one to be used as the measure in a data cube using OLAP. The reason is labor quantities or man-hours generally have a large volume of data and can be aggregated during cube processing. Construction material, as discussed above, may also be designed in a data cube as one kind of measure but must deal with the unit issue carefully. Construction equipment seems not to be suitable for OLAP-based analysis because of the limited quantity involved in projects, except perhaps in certain heavy civil types of projects.

## **CHAPTER 4: A PROPOSED OLAP-BASED PM SYSTEM**

### **4.1. Introduction**

Based on the mapping of the general OLAP concepts to Project Management processes in the construction industry, this chapter discusses a Web-based OLAP-integrated project management system to facilitate the practices in the industry. The chapter introduces a proposed OLAP-based System for Cost Control and Manpower Allocation, which is a preliminary attempt to fulfill all the requirements for integrating OLAP theories with project management. An overview of the proposed prototype, along with the simplifications and basic functionalities, is discussed in this chapter. Also included in this chapter is a small imaginary project with some sample data that will be used for testing and evaluating the proposed prototype.

### **4.2. Overview of an OLAP-based PM Platform**

The dramatic features provided by OLAP contribute one of the critical motivations to consider OLAP as an ideal technology to upgrade the current IT platforms for project management. The new proposed PM platform is intended to be shared among all project participants; thus the system is required to represent data as per various project views from different levels. The basic operations of OLAP—such as drill-through, slice & dice, and rotation—meet the requirements and extend the reach of project data to all levels of a project team. From a project manager who takes charge of the whole project, to a foreman who is only responsible for a labour crew, the end-user can find a way of presenting the data that meets his/her own interests.

The OLAP-based solution should have the capability to present data in a user-specified dynamic manner. Because end-users are from various backgrounds, they may expect different presentations of data. For example, project-oriented data could be represented in terms of multidimensional matrices, which are composed of grids of data, primarily

numbers. This kind of data representation generally satisfies analysis where the emphasis is on trying to analyze quantities or on seeing exact values in detail (such as how much material cost a specific organization spends on a specific day). However, for users who are more concerned with things like finding upper and lower ranges (such as the peak and nadir of on-site manpower allocation), or like data comparisons (such as the distinction between estimated cost and actual cost), their emphasis is now on data trends and performing analysis—which is better presented by graphics instead of spreadsheets.

Generally any front-end application of an OLAP solution has the flexibility to represent complex data in a variety of ways, such as in spreadsheets and charts. Microsoft Office Web Components (OWC), for example, provide a PivotTable component for numerical data presentation and a Chart component for graphic presentation. Those kinds of tools greatly assist end-users with choosing the most appropriate data presentation in their own way. They can minutely analyze numbers presented in data-tables, or apply visible charts to abstract data making complex project-related data easy to understand.

In summary, a well-established OLAP-based solution is a new way and may be an ideal tool for project data presentation and analysis. Among existing technologies in the IT field, OLAP is one of the fastest and most mature ways in terms of meeting the minimum requirements of a multi-dimensional project data presentation. Although in a very coarse and primitive stage, the integration of OLAP technology with project management is promising and may have great potential for further development.

### **4.3. Simplifications of the Proposed Prototype**

Due to time and resource limitation, the prototype developed in this thesis research mainly serves as an initial illustration of certain concepts, rather than a well-established IT platform for direct utilization in the real-world industry. However, the proposed OLAP-based PM system prototype, along with a small sample project, addresses a set of

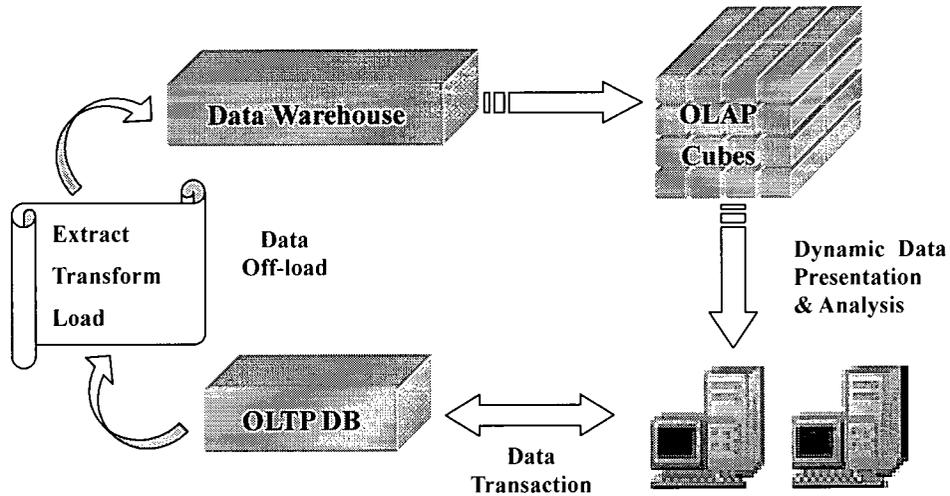
specific project management scenarios—which helps testing the waters to see how the OLAP solution benefits project management and if it does have potential future development space.

Before designing the prototype, several simplifying assumptions were made which are summarized in the following sections.

#### **4.3.1. Data Storage**

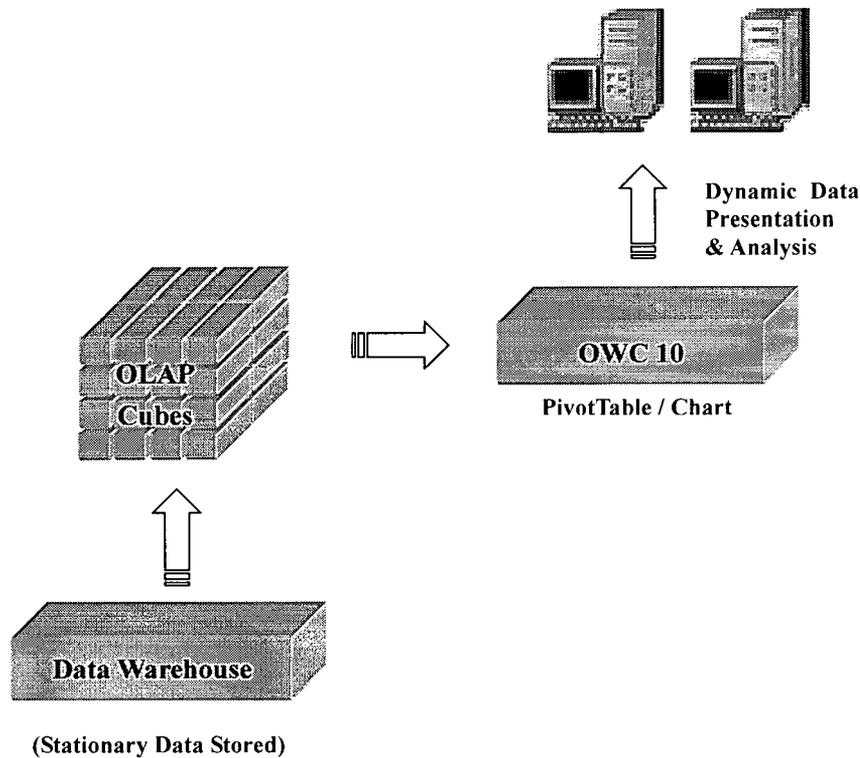
As discussed in chapter 2, a well-rounded OLAP solution generally consists of five main sections: OLTP operational database, ETL services, Data warehouse, OLAP services, and client-end applications. OLTP databases execute real-time transactions, i.e. they are optimized to add, update or delete records during operations. A data warehouse serves as a place for the OLTP database to offload historical data, and handles high volume analytical and reporting queries. The stability and consistency of the historical data stored in the data warehouse enables OLAP to better re-organize data into cubes for rapid analytical queries. The client application serves as a platform for data representation and analysis. Figure 4.1 is a data flow chart for a typical OLAP solution.

There are two modules of design work for a well-rounded OLAP solution. One deals with OLTP database design optimizing the operation of data transaction and temporary storage. The focus is on how to efficiently add, delete, change, and store real-time operational data in the OLTP database, as well as how to offload data to a data warehouse through ETL services (Extract, Transform, Load). Another part of work is about data warehouse and OLAP cube design optimizing data re-organization and presentation for analysis purpose. The focus is on how to define stationary data stored in the data warehouse into data cubes using OLAP technology, as well as on processing the cubes for a clear and dynamic client-side representation.



**Figure 4.1: Data Flow Chart for a Typical OLAP Solution**

Due to research time and scope limitations, this prototype only focuses on the latter, i.e. re-organization and representation of data using OLAP technology. In this case, it was assumed that the sample project data have been previously unloaded from an OLTP database and stored in the data warehouse. The design work thus starts with data warehouse design, followed by OLAP cube design and front-end data presentation design. Figure 4.2 shows the simplified data flow chart for the proposed prototype.



**Figure 4.2: Simplified Data Flow Chart for the Proposed Prototype**

#### **4.3.2. Number of Dimensions**

Although there are various ways in which project data can be organized and accessed, only five selected views (dimensions) were pre-defined in this OLAP-based prototype, including time, participants, task, product, and cost type. This highly simplified the prototype to a manageable scale while still making it reasonable for testing purpose. All of the dimensions selected for cube design are commonly accepted important views in project management, which are of interest to almost all project participants.

#### **4.3.3. Type of Measures**

Two types of data were selected as measures to be plugged into OLAP cubes. One was cost data and the other was human resource data. As for the benchmarks of selecting measures for future cube design, two factors should be carefully considered. One is from

the industrial side that the central data must play an important role in project management processes in correspondence with the selected dimensions. The other thinking is from the technical side as discussed in Chapter 3, that the measures must fulfill all OLAP requirements.

#### **4.3.4. Data Presentation**

The proposed system will use Microsoft Office Web Components (OWC) for data presentation, which may be the fastest way to develop a web-based prototype for testing purposes. In this sample, Microsoft Business Intelligence Portal (BI Portal) was used for cube presentation. Microsoft BI Portal is an integrated, web-based OLAP solution that embeds Microsoft PivotTable and Chart Office Web Components (OWC) into the front webpage. OWC has an identical presentation of Microsoft Office providing fast implementation and easy deployment.<sup>4-1</sup>

#### **4.3.5. Cube Structure**

As discussed in Chapter 2, there are two common types of cube structures: the “star” schema and the “snowflake” schema. In a star schema, each dimension table joins directly to the central fact table, while in a snowflake schema one or more dimension tables do not join to the fact table directly. Instead they join to other dimension tables. The dimension tables that do not connect to the fact table are for dimensions with a complex hierarchy and thus with multiple dimension tables. Considering the degree of database complexity and dimension hierarchies of the sample project, as well as the testing purpose of the prototype, a typical star schema is used during cube design. Although snowflake schemas support ease of dimension maintenance because they are more normalized, star schemas are much easier for direct user access and often support simpler and more efficient queries.

---

<sup>4-1</sup> Detailed introduction of Microsoft BI Portal, along with the implementation environment analysis is discussed in Chapter 6.

#### **4.4. Functionality & Benefits of the Proposed Prototype**

In spite of all the simplifying assumptions made above, the prototype has some basic functions for testing purposes. The capabilities and benefits of the prototype system are discussed in the following sections.

##### **4.4.1. OLAP Integration**

The proposed system is an OLAP-integrated system for dynamic data presentation. As the principal proposal of this research is to apply the OLAP data model concepts to an integrated PM system, the prototype achieves toward this target. A sample data warehouse was developed using imaginary data from a typical construction project. The project-related data stored in the data warehouse was reorganized into data cubes using OLAP technology, and the data can then be presented and analyzed in a dynamic and intuitive way, making the most of the advantages provided by OLAP technology.

##### **4.4.2. Web-based Presentation**

The system is a web-based platform for multiple users. Again, the Unified Project Management (UPM) approach was introduced because of the fact that project participants intend to organize projects from their own perspectives separately, ignoring the hidden interrelations between their specific views. We don't want the new OLAP-integrated PM platform to act as distinct, independent systems operated separately by various project participants. The future OLAP solution, if applicable, should be a web-based platform that can be accessed by multiple users from anywhere with an internet connection.

##### **4.4.3. Friendly UI**

The system is user-friendly, providing clear communication and dynamic presentations. Microsoft Office Web Components (OWC) is used in the presentation layer, which allows project-related data to be represented in PivotTables dynamically in a user-specified way for numerical analysis. Moreover effective, data-driven graphics can communicate high

volumes of project data intuitively at a glance. The Chart component and the PivotTable component can be connected so the data presentation and the graphics change simultaneously when users conduct different meaningful analysis. Users can simply drag and drop dimensions as they want so as to develop various views of data presentation. They can also filter data by any member in any dimension.

#### **4.4.4. Structure Consistency**

The system has a consistent structure with high integration of tools from Microsoft. This ensures much easier future upgrade and deployment. The system architecture is based on the Microsoft .NET platform. The database can be developed using MS Access or MS SQL Server. Data cubes are processed in MS SQL Analysis Services. OWC are embedded in the presentation webpage which have identical interface with MS Office XP. There are no specific deployment requirements on the client side. As long as the end-user has internet connection and Microsoft Office XP installed on the computer, he or she can operate the system (assuming he/she is authorized to access the database).

#### **4.5. Sample Project**

To test and evaluate the proposed OLAP-base PM system, a sample project is used to provide relevant project data. Considering the research scope and time limitation, the project is an imaginary small-sized foundation construction for an office building. It is assumed the project started for bulk earthwork on Jan 6, 2003 and finished on Feb 14, 2003 to deliver the concrete spread foundation with piers and the concrete elevator pit.

The general contractor has a general labour crew, a rough carpentry crew, and a concrete crew taking part in the project. The G.C. is responsible for all formwork, rebar, and concrete but the earthwork, electrical, plumbing and elevator equipment is sub-traded to other contractors. The project target is to deliver the products – the finished concrete foundation and elevator pit—on time and under budget. Some summary information for

the sample project is attached at the end of the chapter.

Besides schedule, participants, tasks, and product information, the imaginary project data also includes detailed costs and man-hour data. The made-up cost values will be used to test the performance of the OLAP-integrated system to do cost tracking and analysis. The made-up man-hour information will be used to test how efficient the OLAP-based system to analysis manpower allocation.

All this information is to be included in the data warehouse and finally in the OLAP cubes for a dynamic representation through an OLAP-oriented user interface. The design of the data warehouse and the OLAP cubes will be discussed in the following chapters.

ACTIVITY DESCRIPTION	DUR.	START	FINISH	06-Jan-03	07-Jan-03	08-Jan-03	09-Jan-03	10-Jan-03	11-Jan-03	12-Jan-03	13-Jan-03	14-Jan-03	15-Jan-03	16-Jan-03	17-Jan-03	18-Jan-03	19-Jan-03	20-Jan-03	21-Jan-03	22-Jan-03	23-Jan-03	24-Jan-03	25-Jan-03	26-Jan-03	27-Jan-03	28-Jan-03	29-Jan-03	30-Jan-03	31-Jan-03	01-Feb-03	02-Feb-03	03-Feb-03	04-Feb-03	05-Feb-03	06-Feb-03	07-Feb-03	08-Feb-03	09-Feb-03	10-Feb-03	11-Feb-03	12-Feb-03	13-Feb-03	14-Feb-03	15-Feb-03	16-Feb-03						
				m	t	w	t	f	s	s	m	t	w	t	f	s	s	m	t	w	t	f	s	s	m	t	w	t	f	s	s	m	t	w	t	f	s	s	m	t	w	t	f	s	s						
Excavate foundations	10 days	06-Jan-03	17-Jan-03	+	+	+	+	+			+	+	+	+	+																																				
Excavate elevator pit	2 days	06-Jan-03	07-Jan-03	x	x																																														
Form column piers and spread foundations	4 days	16-Jan-03	21-Jan-03											+	+			+	+																																
Rough-in electric and plumbing in elevator	4 days	08-Jan-03	13-Jan-03			x	x	x			x																																								
Form elevator pit walls	4 days	14-Jan-03	17-Jan-03									x	x	x	x																																				
Set reinforcing and anchor bolts	4 days	21-Jan-03	24-Jan-03																+	+	+	+																													
Pour column piers and foundations	5 days	27-Jan-03	31-Jan-03																								+	+	+	+	+																				
Pour concrete elevator walls	1 day	20-Jan-03	20-Jan-03															x																																	
Cure elevator wall concrete	7 days	21-Jan-03	29-Jan-03																x	x	x	x				x	x	x																							
Cure piers and foundations	7 days	03-Feb-03	11-Feb-03																																																
Strip wall forms	1 days	30-Jan-03	30-Jan-03																																																
Strip column piers and foundation forms	3 days	12-Feb-03	14-Feb-03																																																
Install pneumatic tube in elevator pit	3 days	31-Jan-03	04-Feb-03																																																
Prepare and pour concrete floor in elevator pit	1 day	05-Feb-03	05-Feb-03																																																

ACTIVITY DESCRIPTION	PARTICIPANTS	LEGEND	PRODUCT CATEGORIES	PRODUCT / DELIVERABLES
Excavate foundations	Site Grading Contractor	+++++++	Piers & Spread Foundation	Foundation Excavation
Form column piers and spread foundations	G.C. Rough Carpenter Crew		Piers & Spread Foundation	Ready for Concrete Pouring (Foundation)
Set reinforcing and anchor bolts	G.C. Rough Carpenter Crew		Piers & Spread Foundation	Ready for Concrete Pouring (Foundation)
Pour column piers and foundations	G.C. Concrete Crew		Piers & Spread Foundation	Finished Concrete Foundation & Piers
Cure piers and foundations	G.C. General Labor Crew		Piers & Spread Foundation	Finished Concrete Foundation & Piers
Strip column piers and foundation forms	G.C. General Labor Crew		Piers & Spread Foundation	Finished Concrete Foundation & Piers
Excavate elevator pit	Site Grading Contractor	xxxxxxxx	Elevator Pit	Elevator Pit Excavation
Rough-in electric and plumbing in elevator	Plumbing Contractor, Electrical Contractor		Elevator Pit	Ready for Concrete Pouring (Elevator Walls)
Form elevator pit walls	G.C. Rough Carpenter Crew		Elevator Pit	Ready for Concrete Pouring (Elevator Walls)
Pour concrete elevator walls	G.C. Concrete Crew		Elevator Pit	Finished Concrete Elevator Walls
Cure elevator wall concrete	G.C. General Labor Crew		Elevator Pit	Finished Concrete Elevator Walls
Strip wall forms	G.C. General Labor Crew		Elevator Pit	Finished Concrete Elevator Walls
Install pneumatic tube in elevator pit	Elevator Contractor		Elevator Pit	Finished Elevator Pit
Prepare and pour concrete floor in elevator pit	G.C. Concrete Crew		Elevator Pit	Finished Elevator Pit

## **CHAPTER 5: DATA WAREHOUSE MODEL AND OLAP CUBE DESIGN**

### **5.1. Introduction**

This chapter addresses data warehouse modeling, as well as the design of two OLAP cubes to test and evaluate the prototype. Microsoft SQL Server Analysis Manager is used to build and process OLAP cubes. A foundation construction project is used as a sample project to provide project data, which is briefly introduced at the end of the chapter.

In this chapter, most of the introduction material—the direct quotations, the general description and some definitions—relating to data warehouse design are developed from the chapter 17 of Microsoft SQL Server 2000 Resource Kit—“Data Warehouse Considerations” [Browning, 14].

### **5.2. Data Warehouse Design**

#### **5.2.1. Dimensional Model Design**

As discussed in the foregoing chapter, a data warehouse provides a place for an online transaction processing (OLTP) database to offload data. Typically OLTP operational databases execute daily transactions—they are optimized to store, update, and edit the data, not to help users analyze it. On the contrary, a data warehouse is a specialized database designed to store and re-organize static data. It is an optimized relational database that supports OLAP cubes to provide instantaneous query responses for analysis in a more efficient way. Although in most cases both a data warehouse and an OLTP operational database are relational databases, they may have different structural models because the purpose of a data warehouse generally differs from that of an OLTP operational database. Table 5.1 [Browning, 14] shows the comparison of a data warehouse database and an OLTP operational database.

**Table 5.1: OLTP Operational Database vs. Data Warehouse Database [Browning, 14]**

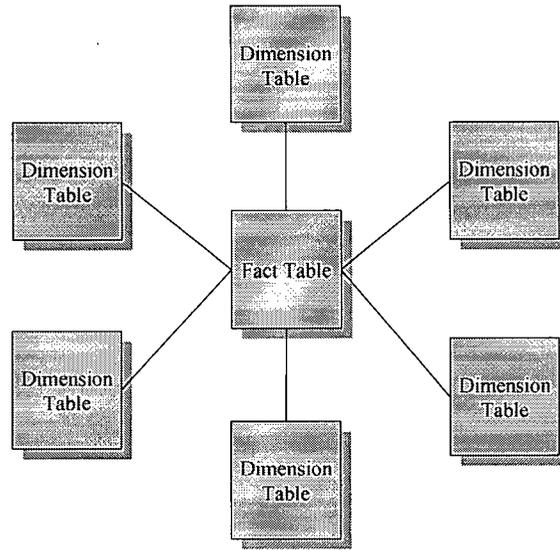
<b>OLTP Operational Database</b>	<b>Data Warehouse Database</b>
Designed for real-time data transactions	Designed for dynamic data analysis
Constantly changing data; Users perform data-entry and editing tasks	Loaded with consistent, valid data; users never perform data-entry and editing tasks
Generally contains a large number of tables with complex interrelationships	Use much fewer tables with a different type of schema ("star" or "snowflake")
Optimized for a common set of transactions, usually dealing with a single row per table at one time	Optimized for bulk loads and complex analytical queries that access many rows per table

An OLTP database requires a more “normalized structure in order to provide validation of input data, minimize redundancy, and support a high volume of fast transactions”. Thus its structural model often looks like “a wide spider web of hundreds of tables interrelated with each other”. On the contrary, a data warehouse requires a “typical dimensional model that supports simplified analytical queries and provides outstanding query performance by minimizing table joins”. When realized in a database, the dimensional model for a data warehouse may produce either a star schema or a snowflake schema that contains a central fact table and multiple dimension tables surrounding it. In a star schema, all dimension tables join directly to the central fact table while in snowflake schema one or more dimension tables join the fact table through other dimension tables.

By establishing dimension hierarchical levels from separate tables, snowflake schemas support ease of dimension maintenance since they are more normalized, but when dimension hierarchies are not complex, star schemas are more straightforward and support simpler and more efficient analytical queries. Considering the complexity of the dimension hierarchies of the sample project, a star schema was used for the data warehouse.

Figure 5.1 is a star schema diagram. There is a central fact table with several dimension tables connecting with it directly. The fact table contains real information of event details

for summarization. Because dimension tables contain records that describe facts, the fact table can be reduced to columns for dimension foreign keys and numeric fact values. Detailed fact table design, dimension table design and overall structural schema are discussed in the following sections.



**Figure 5.1: Data Warehouse Star Schema Diagram**

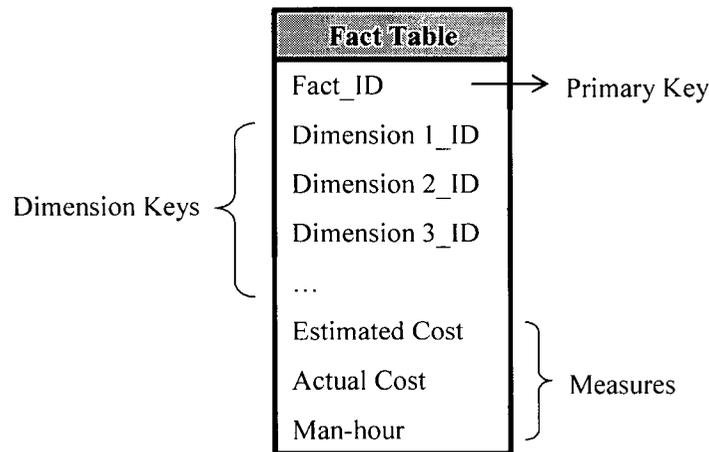
### **5.2.2. Fact Table Design**

The fact table contains real information of event details for summarization. It is supposed to address project processes or problems that need to be analyzed by end-users. The numeric data stored in the fact table, often referred to as “measures”, represent the project processes and quantify the facts. For the sample projects, measures were selected to represent two predominant industrial focuses—cost control and manpower allocation. Thus two kinds of measures are selected accordingly—construction costs and man-hours, both of which are numeric and additive.

Both “estimated costs” and “actual costs” were included in the fact table. Simultaneously presenting estimated costs and actual costs gives the end-users a straightforward view that helps them better track construction cost. Users will be able to review the differences between estimated costs and actual costs in a dynamic and intuitive manner. Man-hour

measure stored in the fact table results in a display of manpower allocation through OLAP front-end applications. It provides the site manager or superintendent with manpower distribution by time, by task, by participants, or any other possible dimensions pre-defined.

Besides measures, which are numeric fact values and will be pre-computed during OLAP cube processing, the fact table also contains foreign key columns that link with primary keys of each dimension table. The fact table actually resolves “many-to-many relationships between dimension tables because dimension tables join with each other through the central fact table”. Figure 5.2 shows the fact table structure.



**Figure 5.2: Data Warehouse Fact Table Structure**

### 5.2.3. Dimension Table Design

Unlike the fact table, the dimension tables don't contain real fact values. Instead they summarize the attributes associated with fact values and organize these attributes into distinct views such as time, tasks, products, etc. Those attributes basically serve as query constraints in OLAP solutions. Because dimension tables don't contain fact values, they are usually very small.

As per users' analytical needs, the data stored in dimension tables is generally hierarchical. The hierarchy enables OLAP-based data summarization, drill-through, and

“slice & dice” operations, which provide dynamic data presentation for analysis. The hierarchy is actually fulfilled during cube design and processing, but it is important to take consideration of the hierarchy issue as early as starting dimension table design.

To develop the data warehouse structure for the sample project, five dimensions were included which contain time, tasks, products, participants, and cost types.

- **Time Dimension**

The time dimension provides an effective mechanism for time-oriented analysis. Suppose the imaginary OLTP database is organized and updated on a daily basis, and then the time dimension table in the data warehouse should also adopt a daily time granularity. It was organized into four levels during cube processing: “year”, “quarter”, “month”, and “date”.

- **Task Dimension**

The task dimension provides a process-based view of the project that splits the entire project into activities (or tasks). This dimension was organized into two levels during cube processing—“task category” and “task”. Based on the sample project, all activities were grouped into four different task categories: “formwork”, “rebar”, “concrete” and “installation”. In terms of OLAP technology and terminology, those four task categories were referred as four “members” of the task dimension in the task category level.

- **Participant Dimension**

The participant dimension provides an organizational view of the project indicating who, or which organizations, crews, or personnel, participate in the project. As with the task dimension, the hierarchy of the participant dimension included two levels during cube processing, which were “participant category” and “the participant”. Based on the sample project, all participants were organized

into two distinct participant categories, “general contractor” and “sub contractor”.

- **Product Dimension**

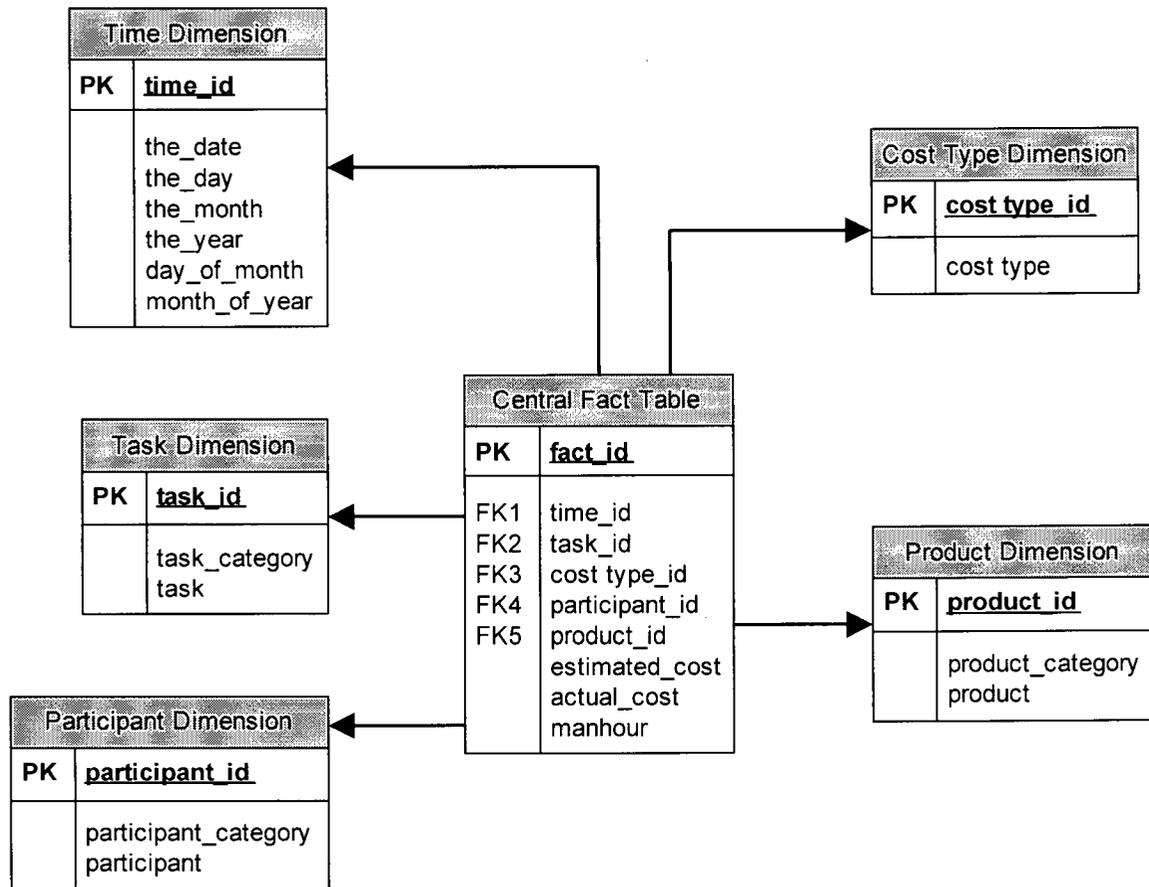
The product dimension provides a product/deliverable view of the project indicating artefacts that can be delivered at a specific period of time. For this sample project, the product dimension had two members—“piers & spread foundation” and “elevator pit”.

- **Cost Type Dimension**

Because construction cost is selected as an OLAP cube measure, a cost type dimension is necessary for a better and more detailed cost control. The hierarchy of this dimension had only one level. The three basic cost types, i.e. “labour cost”, “material cost” and “equipment cost”, constituted this dimension.

#### **5.2.4. Multidimensional Structure**

As mentioned, a star schema was adopted for the data warehouse. Figure 5.3 shows the whole multidimensional structure of the data warehouse in a star schema.



**Figure 5.3: Data Warehouse Structure – “Star” Schema**

### 5.2.5. Sample Project Data

The sample data plugged into the data warehouse is based on the “Foundation Construction” sample project introduced in the previous chapter. Again, for a well-rounded OLAP-based PM system, the data stored in the data warehouse should be unloaded from OLTP databases through ETL. However, due to the research scope and time limitation, the sample project data were plugged into the data warehouse directly as if they were unloaded from the real-time OLTP databases. This simplification and assumption has been discussed in the “simplifications” section of the last chapter.

### **5.3. OLAP Cube Design & Pre-processing**

#### **5.3.1. Overview**

The OLAP cubes are not designed to store data or to optimize data transactions. The main function of the OLAP cubes is to provide high performance in responding to complex analytical queries by pre-processing and summarizing the stable and consistent historical data stored in the data warehouse.

A data warehouse provides a multidimensional view of data in an intuitive model, generally using a star schema or a snowflake schema designed for dynamic and complicated analytical queries. OLAP organizes data warehouse data into multidimensional data cubes based on this dimensional model, and then pre-processes these cubes to provide maximum performance for queries that summarize data in various ways. In Microsoft SQL Server 2000, the Analysis Services provides tools for OLAP cube design and processing.

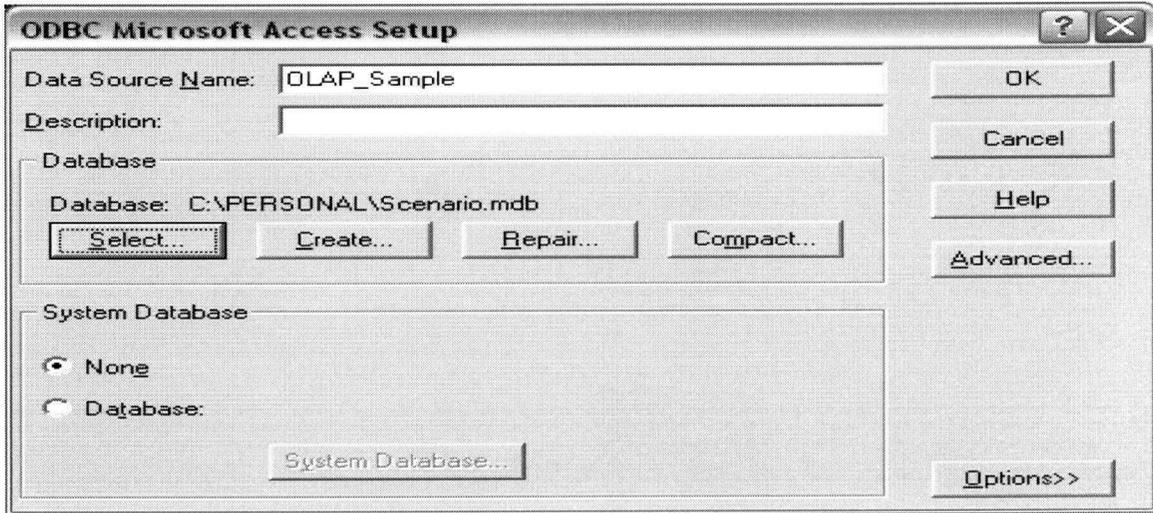
Two OLAP data cubes were developed in accordance with the measures stored in the fact table of the data warehouse. One cube is for cost control and the other is for man-power allocation.

Note the dimensions discussed in the forgoing section are all shared dimensions, which are in use for both of the proposed cubes. Shared dimensions offer a performance advantage because they make more efficient use of memory. This allows for faster response times. In very rare cases, for example when a high level of security for sensitive data is necessary, a private dimension can be used in one specific cube and be inaccessible to other cubes. In most cases, however, it is much better to use shared dimensions.

#### **5.3.2. Set Up Data Source Connections**

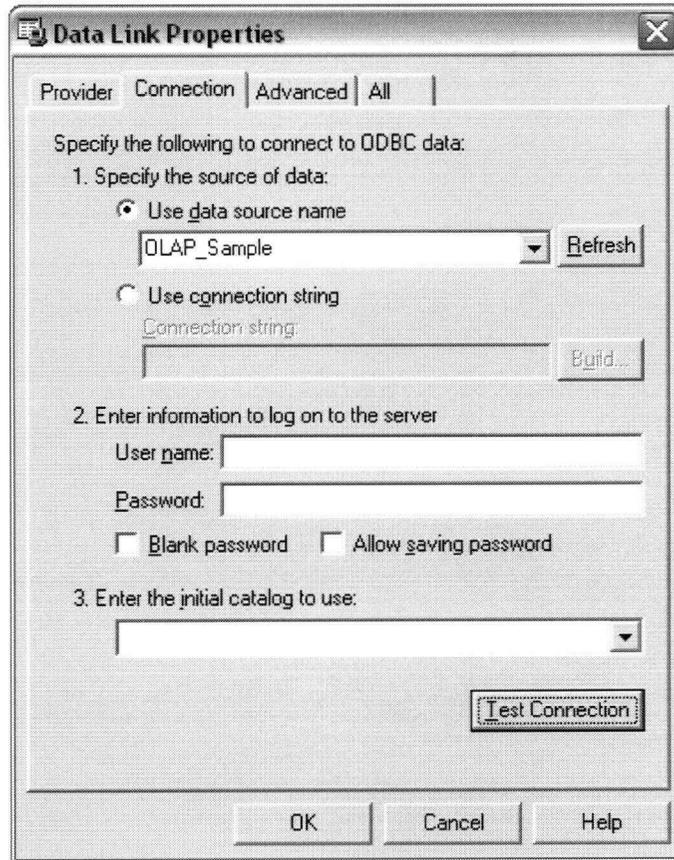
The first step was to set up a new system data source name (DSN) through the

“Administrative Tools” from the Control Panel. A new data warehouse was created and named “OLAP\_Sample” (Figure 5.4).



**Figure 5.4: Set Up System Data Source Name**

Next Microsoft SQL Analysis Manager was opened to set up the database structure. In the Analysis Manager Tree pane, after right-clicking the Data Sources folder under the “OLAP\_Sample” database, the “New Data Source” was chosen to establish data source connection. On the “Provider” tab of the Data Link Properties dialog box, the “Microsoft OLE DB Provider for ODBC Drivers” option was selected. Then from the “Use data source name” list, the “OLAP\_Sample” data source was selected (Figure 5.5). When clicking the “Test Connection” button, a message box opened saying that the data source connection was successfully completed. By then it was ready for the next step to build OLAP data cubes using Microsoft SQL Analysis Manager.

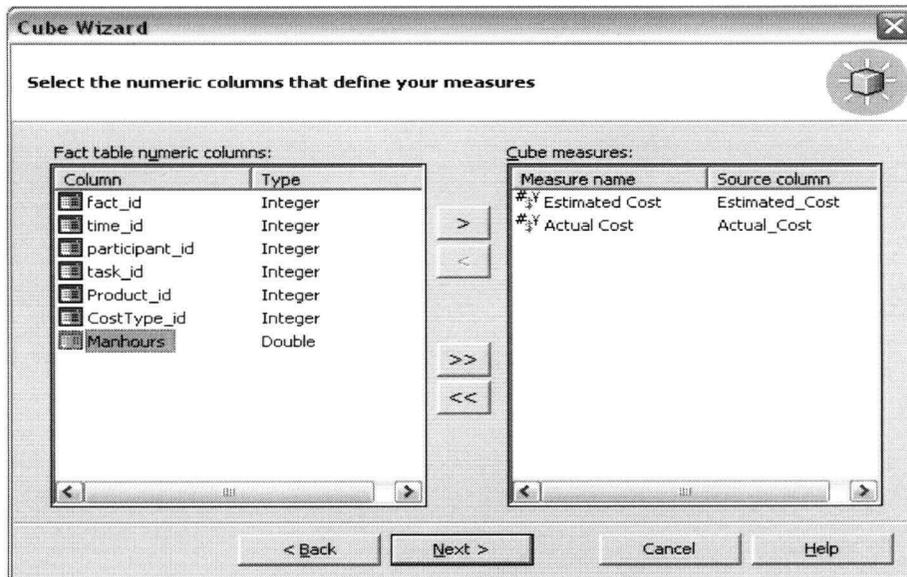


**Figure 5.5: Make Data Source Connection**

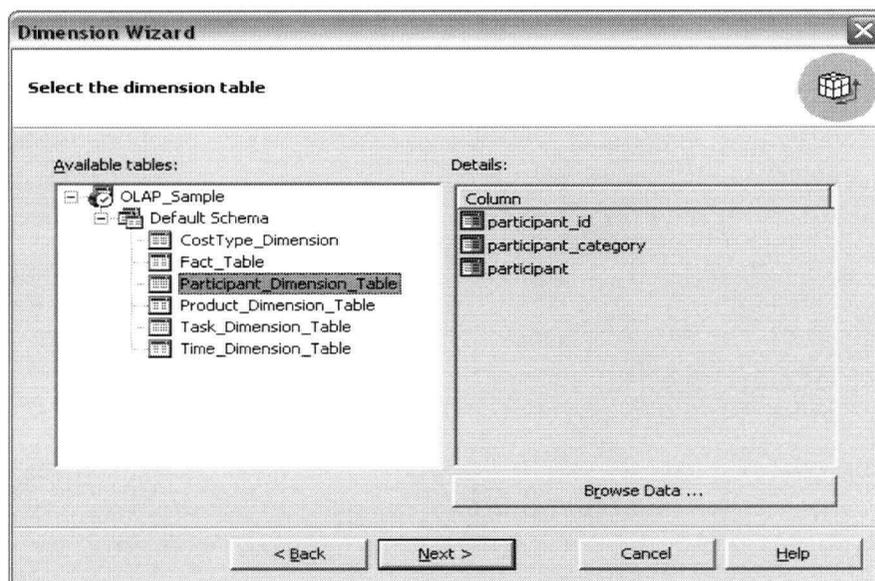
### **5.3.3. Build OLAP Data Cubes**

Two OLAP data cubes were created—a “Cost Control” cube and a “Manpower Allocation” cube. To build the “Cost Control” data cube, the Cube Wizard of the Analysis Manager was opened, and then both the Estimated Cost and the Actual Cost from the fact table were selected as the cube measures (Figure 5.6). To build the “Manpower Allocation” data cube, the same procedure was repeated, but the Manhour from the fact table was selected as the cube measure.

In total, there were five shared dimensions defined for each cube, including Time, Task, Participant, Product, and Cost Type. The members and hierarchies of each dimension were defined by selecting them from the data warehouse dimension tables through the Dimension Wizard in Analysis Manager (Figure 5.7).



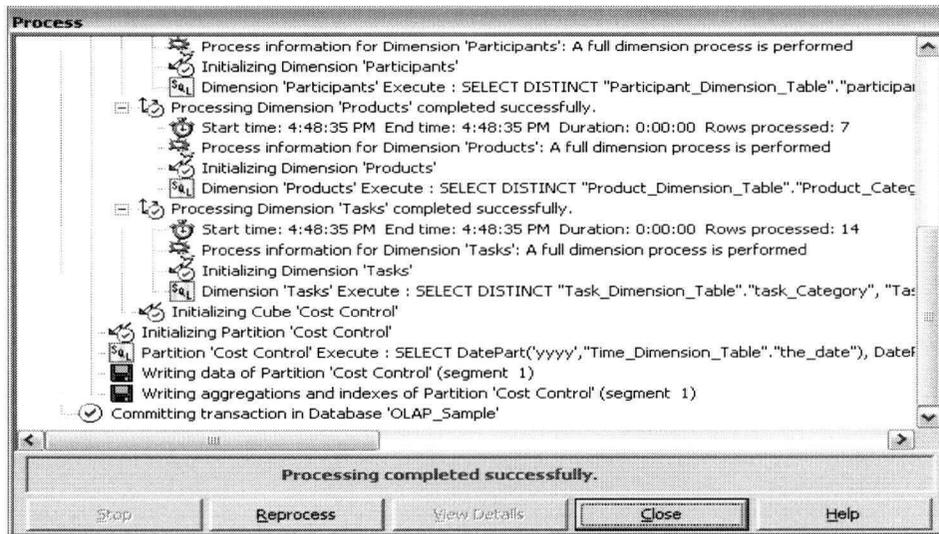
**Figure 5.6: Define Cube Measures**



**Figure 5.7: Define Cube Dimensions**

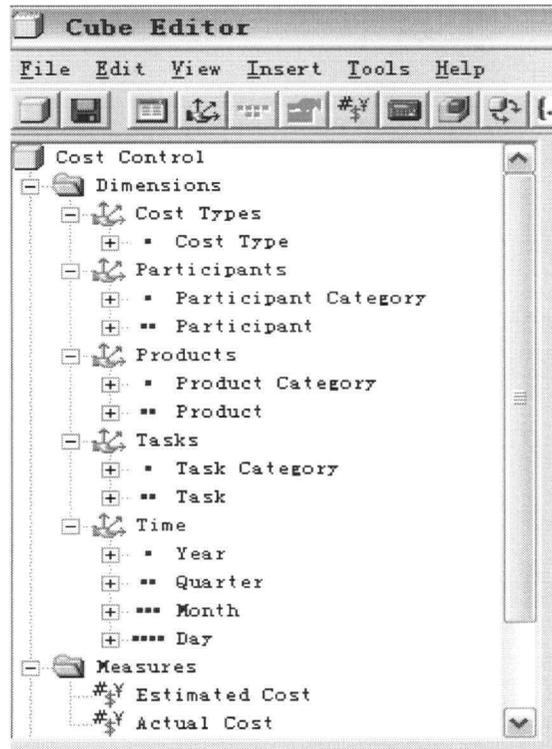
After defining the cube measures and dimensions, the cube storage was designed and the cubes were processed (Figure 5.8). Microsoft SQL Server 2000 Analysis Services allows the users to set up aggregations—the pre-calculated summaries of data that greatly improve the efficiency and response time of queries. When processing a cube, the aggregations designed for the cube are calculated and the cube is loaded with the

calculated aggregations and data.



**Figure 5.8: Processing Data Cubes in MS SQL Server Analysis Services**

After being processed, the data cube can be displayed and edited from the Cube Editor. Figure 5.9 shows the summary of the “Cost Control” data cube from the Cube Editor window. It has five dimensions with hierarchies and two measures (“Estimated Cost” and “Actual Cost”). The “Manpower Allocation” data cube has the same shared dimensions but only one measure (“Manhours”).



**Figure 5.9: Summary of “Cost Control” Data Cube**

## CHAPTER 6: WEB-BASED DATA PRESENTATION

### 6.1. Introduction

This chapter provides a brief introduction of relevant technologies and tools that are used to test the prototype. Microsoft .NET and Web Services provide the basic platform for development; Microsoft BI Portal serves in the prototype as the end-user application for OLAP data presentation; Microsoft OWC provides the main components integrated with the Microsoft BI Portal; and Microsoft SQL Server supports the OLAP data cube design and organization. All these topics are discussed briefly in this chapter.

In this chapter, most of the introduction material about Microsoft's technologies and tools is developed or quoted from contents on the Microsoft® (or MSDN) website.

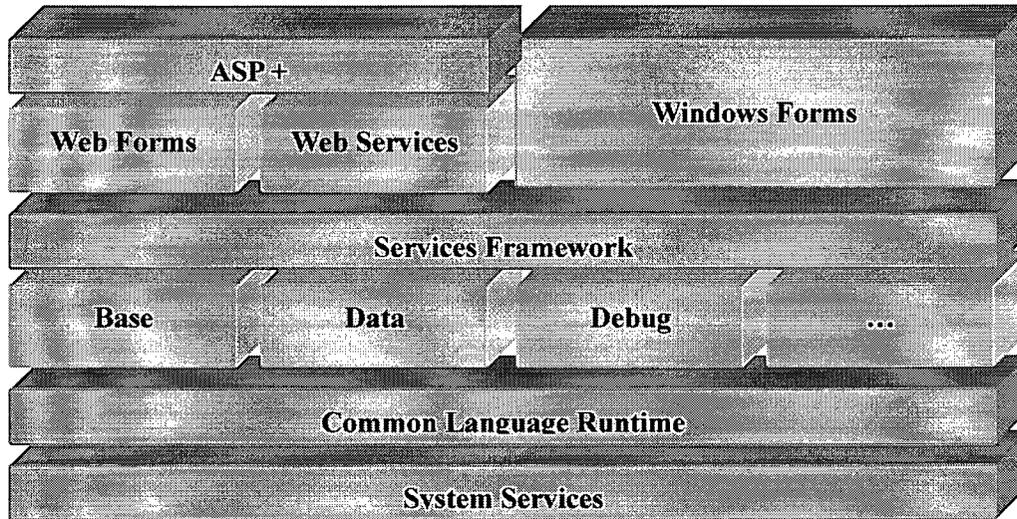
### 6.2. Microsoft .NET and Web Services

Microsoft .NET is the Microsoft strategy for connecting information and applications through an Internet environment. With the help of .NET technology, people can access, manage, and deploy distinct and disconnected solutions quickly, and thus communicate and collaborate more effectively.

The .NET technology helps workstations from different operating system platforms work together by exchanging messages through Web services. Web services are small building blocks based on "industry protocols that include XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), and WSDL (Web Services Description Language). These protocols help computers work together across IT platforms and programming languages" [Microsoft, 22].

The .NET Framework provides comprehensive IT platforms that offer fast and effective ways to build Web services. "With the .NET Framework, developers can take advantage of a programming model designed from the ground up for creating Web services in a

highly productive, multi-language environment” [Microsoft, 22].



**Figure 6.1: Microsoft .NET Framework Architecture [Microsoft, 22]**

Figure 6.1 [Kirtland, 23] shows the Microsoft .NET Framework architecture. The common language runtime, which sits on top of the operating system services, manages the requirements of running code developed in the .NET programming languages. Above the common language runtime is the Services Framework, which provides a set of classes that can be called from the programming language. Conceptually there are two application models sitting on top of the Services Framework—the Web-based application and the Windows-based application, among which the Web-based applications is built on the Microsoft .NET Framework [Kirtland, 23].

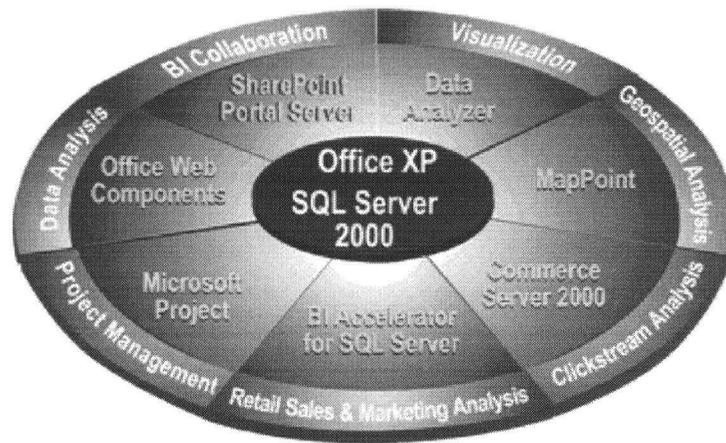
### **6.3. Microsoft Business Intelligence Platform**

#### **6.3.1. Overview**

Business Intelligence (BI), as a technology for processing large volumes of data, provides the foundation for end-to-end solutions with both sophisticated data management and easy-to-use analysis and reporting tools [Larsen, 24]. It can greatly facilitate data-oriented analytical decision making by organizing data much more efficiently and presenting data

in a more dynamic and manageable manner. The Microsoft BI Platform is made up of two major components, Microsoft SQL Server (with Analysis Services) and Microsoft Office System. The SQL Server supports database management and data manipulation that provides an analytics engine in the back-end, while the Microsoft Office system offers user interface in the front-end.

Microsoft BI Platform provides the building blocks for delivering solutions, which eliminate the need for highly specialized software and implementation skills. Through a variety of front-end tools, such as Microsoft Office that are familiar to most users, Microsoft BI platform enables almost all users to dig into the data, manipulate and better understand it, and make the most of it. Figure 6.2 [Microsoft, 25] is the diagram provided by Microsoft showing the platform for Business Intelligence.



**Figure 6.2: The Microsoft® Platform for Business Intelligence [Microsoft, 25]**

### 6.3.2. Microsoft SQL Server 2000

Microsoft SQL Server 2000 provides several tools for data storage and analysis, which include a relational database engine for storing data, the data transformation services for unloading data, and the analysis services for data analysis.

- Analysis Services:

Microsoft SQL Server 2000 Analysis Services provides a set of integrated

analysis services including data-mining features and an Online Analytical Processing (OLAP) engine. It enables fast data access by creating multidimensional OLAP cubes from information stored in the data warehouse. The Analysis Services can work well on large volumes of data. It offers sophisticated analytical capabilities that enable users to act immediately on analytical insights, and enables the application to respond quickly to dynamic conditions [Larsen, 24].

- **Data Transformation Services:**

Microsoft SQL Server 2000 Data Transformation Services (DTS) provides a set of tools that lets the user extract, transform, and consolidate data from disparate sources into single or multiple destinations [Larsen, 26]. It provides the tools to consolidate data from diverse operational systems and databases, an OLTP database for example, into the data warehouse or data marts that support better analysis and decision making enhanced by Analysis Services and OLAP. DTS can perform at pre-determined intervals or in response to specified events.

- **Reporting Services:**

Microsoft SQL Server Reporting Services enables the creation, management, and delivery of both traditional, paper-oriented reports and interactive, Web-based reports [Microsoft, 27]. Being an integrated part of the Microsoft business intelligence framework, Reporting Services supports a wide range of data sources, as well as multiple data output formats such as common Web browsers and Microsoft Office System applications.

### **6.3.3. Microsoft Office System**

The Microsoft Office system is a set of familiar programs that are installed on millions of computers worldwide. By using Microsoft Office to perform sophisticated analysis against large volumes of data, a business intelligence solution extends its reach to more

people, benefiting from users' familiarity with the Office system.

The Microsoft Office XP tools are completely integrated with SQL Server Analysis Services, either directly or over Internet connections. Some of the tools are for data analysis, and some are for data visualization.

- Data Analysis Tools:

Integrated with the business intelligence platform, Microsoft Excel is a calculation engine and data analysis tool combined with charting and query features that enable users to access, analyze, create, and share business data. With Excel PivotTables and PivotCharts, users can rotate rows and columns to develop different data summaries, and filter the source data (known as "data slice") to display the specific details review and analysis.

As discussed before, Microsoft Office Web Components are "a collection of Component Object Model (COM) controls, shipped with the Microsoft Office package, for publishing spreadsheets, charts, and databases to the Web" [Microsoft, 28]. They extend the power of the desktop-based Excel to web-based business intelligence applications that allow users to manipulate data from the Web browser. The PivotTable and Chart components are especially useful for data presentation in business intelligence solutions, because they are designed to deliver interactive data reporting and analysis. By connecting them to a pre-defined OLAP server cube, users can view data grouped, sliced, and sorted in a variety of ways, and create dynamic reports for interactive data analysis.

- Data Visualization Tools:

Some multi-dimensional information inherent in business intelligence is very difficult to read and understand unless the data are presented graphically. In these cases, visualization tools are important to help conduct meaningful analysis, make comparisons, and spot trends, making data analysis more understandable and

intuitive. Microsoft Data Analyzer and Microsoft MapPoint are two Office applications that combine powerful data analysis features with data visualization. Both of them allow users to connect to Microsoft SQL Server Analysis Services, and conduct analysis and visualization of data from the desktop.

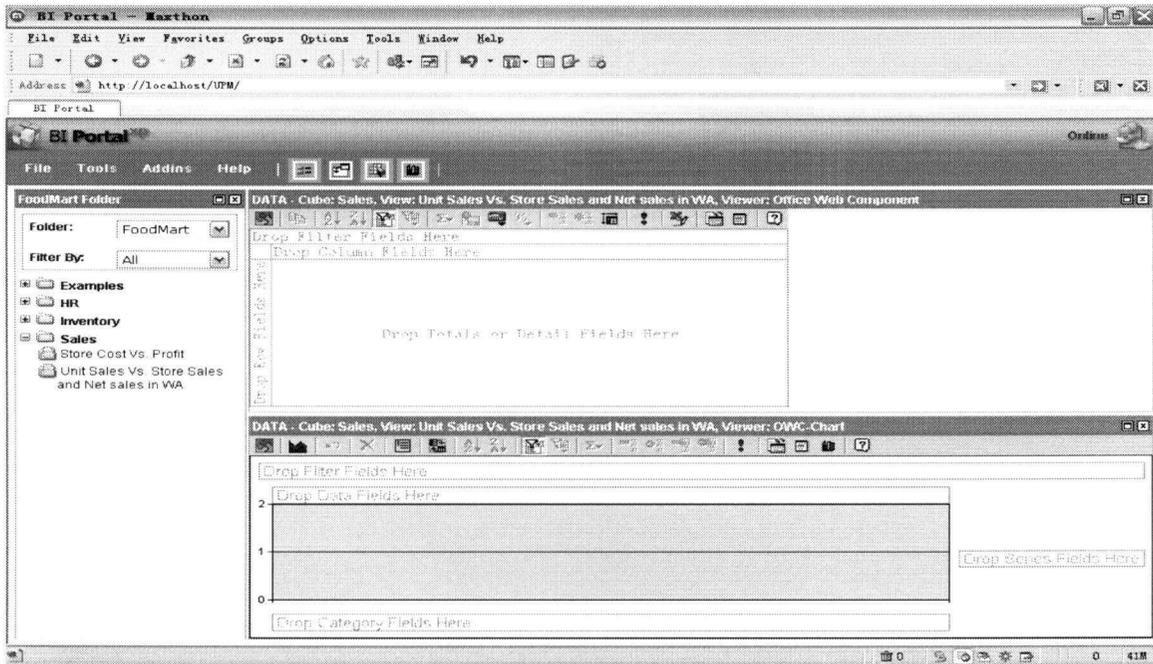
## **6.4. Microsoft BI Portal**

### **6.4.1. Overview**

*(The introduction material in this section (6.4 Microsoft BI Portal) is mainly developed or quoted from “Microsoft Business Intelligence Portal—BIP” [Ben-Moshe, 29].)*

The Microsoft Business Intelligence Portal (BI Portal) is an integrated, Web-based Online Analytical Processing (OLAP) solution that allows OLAP cube data to be dynamically presented in multi-dimensional views. Microsoft OLAP client tools, such as Microsoft PivotTable and Chart Office Web Components (OWC), are integrated into BI Portal to provide a complete OLAP solution platform. BI Portal was used to test the sample data warehouse and data cubes developed for the prototype.

BI Portal is free and can be downloaded from Microsoft’s website. Being a .NET web-based application, it can be loaded by directing a web browser to the web server on which it was installed. Figure 6.3 shows the main user interface of the Microsoft BI Portal.



**Figure 6.3: Microsoft BI Portal User Interface**

The main user interface of the MS BI Portal is divided into several rectangular screen areas—the Menu bar & Toolbar, the Folder list and, most importantly, the OWC area. The Menu bar and Tool bar provide Windows-like functions, which display user-dependant menu and control items. The Folder list helps users save created views by scenario. The OWC area actually consists of two main components (as discussed in Chapter 2—the OWC PivotTable control and the OWC Chart control). This area provides a rich, integrated graphic and PivotTable display.

#### **6.4.2. BI Portal Architecture**

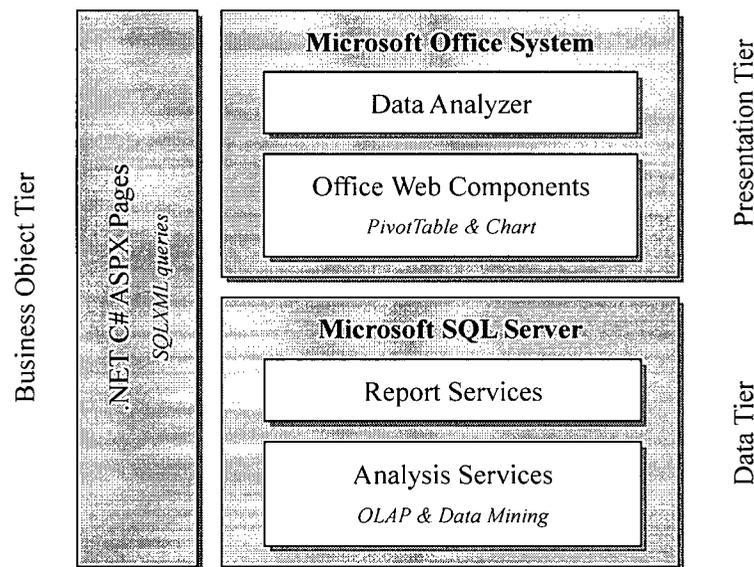
The Microsoft BI Portal is a .NET-based Web application built up in the traditional three-tier model.

In the presentation layer, the Office Web Components serve as the main viewer controls for presenting and manipulating the OLAP cube data in a dynamic and user-specified manner. The Business Object Layer connects the presentation layer to the database using SQLXML queries. The .NET C# ASPX pages and classes provide the presentation layer

with data to get MDX queries for each view from the database.

The Data Layer “uses Data Source objects to encapsulate the technical properties of a data source type” [Ben-Moshe, 29]. Data is presented in the same way using the PivotTable and Chart Office Web Components, no matter whether the data source is an OLAP online cube or an offline cube file. Every Data Source contains the server name, database name, and cube (or table) name so that each data source is uniquely identified. When a data cube is selected by the user, the multidimensional views are fed with the pointed data source to connect to and the query to execute and the OWCs display the received data directly.

Figure 6.4 [Microsoft, 30] shows the Microsoft BI Portal architecture.



**Figure 6.4: Microsoft BI Portal Architecture [Microsoft, 30]**

### 6.4.3. BI Portal Installation and Set-up

Microsoft BI Portal is easy to install and deploy both on the data server and on the client’s side. MS SQL Server 2000 with Analysis Services must be installed on the data server. On the client PC, the only pre-requisites are MS Internet Explorer 6, to browse web pages, and MS Office XP in order to work with Office Web Components in the main

user interface. The Microsoft .NET Framework and SQLXML 3.0 must also be installed on the application server before MS BI Portal can run properly.

During installation of the MS BI Portal, users will be required to select SQL Server and insert the database name (default = "BIP", for this prototype the database name = "UPM"). Then the security mode must be selected between Windows Authentication and SQL Authentication. A user name and password are necessary if SQL authentication is selected. After installing the BI Portal, it can be loaded by opening Internet Explorer and typing (in this case) <http://localhost/UPM> in the URL address box. The Internet Explorer then connects to the Analysis Server and loads the main UI of Microsoft BI Portal.

## CHAPTER 7: PROTOTYPE TESTING AND EVALUATION

### 7.1. Introduction

This chapter describes the testing and evaluation of the proposed prototype based on the warehouse and OLAP cubes created in Chapter 5. The Microsoft BI Portal is used as the front-end OLAP solution for testing purposes. The testing and evaluation targets two project management areas—construction cost control and manpower allocation analysis. Various project management scenarios are developed to support the prototype testing and evaluation.

The objective of testing the prototype is to see whether project management (specifically, cost control and manpower allocation) can benefit from the OLAP technology features introduced in the preceding chapters. The testing should evaluate the practical value of the OLAP in improving project data presentation. To carry out this evaluation, the prototype testing was conducted to combine the OLAP features (such as data drill-through, data slicing and data dicing) with realistic project management scenarios.

To “drill through” an OLAP cube is to navigate from one data item to a set of related items. This “drilling” typically involves navigating up and down through the pre-defined levels in a hierarchy. When selecting data, users can expand or collapse a view of data by drilling down or up in a hierarchy respectively. Data “Slice & Dice” is an informal term referring to data retrieval and manipulation. In an imaginary data cube, each axis of the cube represents a dimension. To “slice” the data is to retrieve a piece (a slice) of the cube by specifying the dimension value from one or more of the dimensions. In effect, the specified dimension values serve as filter criteria in the slicing process. To “dice” the data is to combine pre-defined dimensions in various ways against a two-dimensional matrix, just as if the data cube had been diced into many smaller pieces.

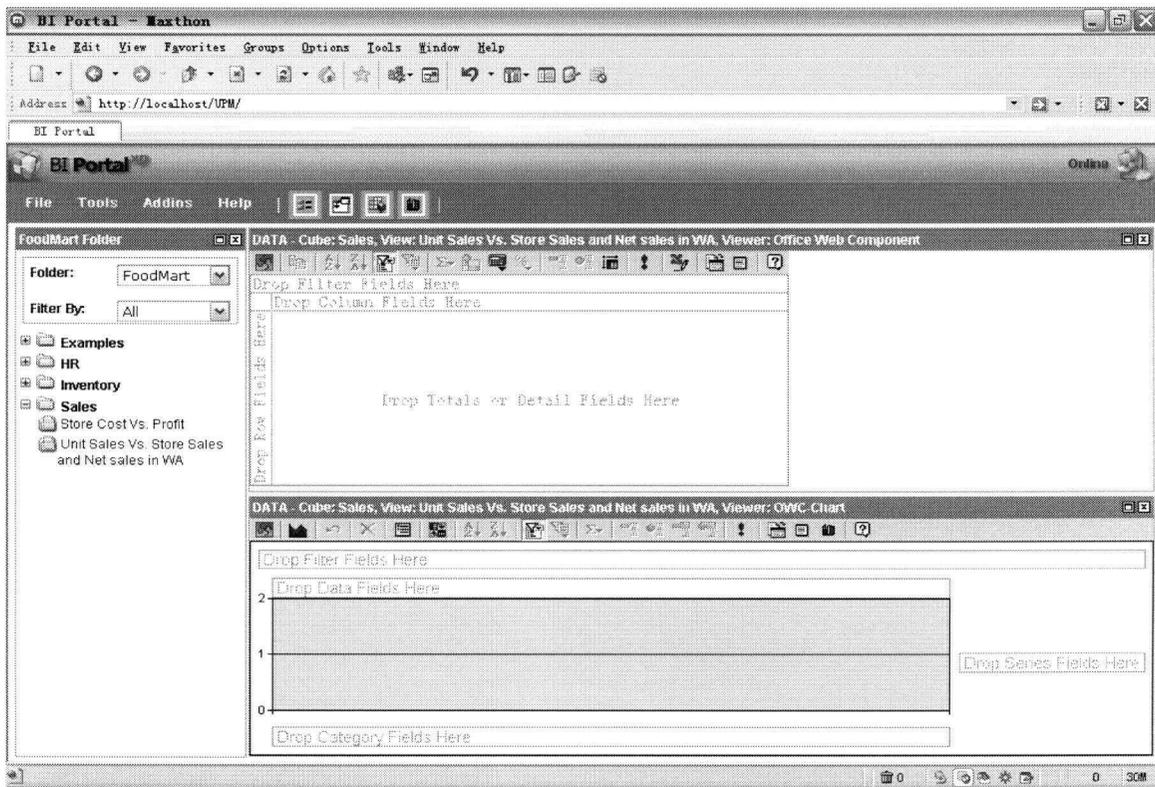
Through several project management scenarios, the tests described in this chapter explore

how the PM process can benefit from the navigation performance provided by OLAP's drill-through and slice & dice features.

## 7.2. Pre-testing Preparation

The prototype testing was conducted on one computer that served as the data server, the application server, and the client. Typically, these three roles would be performed by three different computers, but a single computer was sufficient for testing purposes.

After being installed on a server, the BI Portal was loaded from URL <http://localhost/UPM>, and the main user interface appeared (Figure 7.1). By default, the "Folder" part of the UI referred to the "Food Market" sample database provided with MS SQL Server 2000.

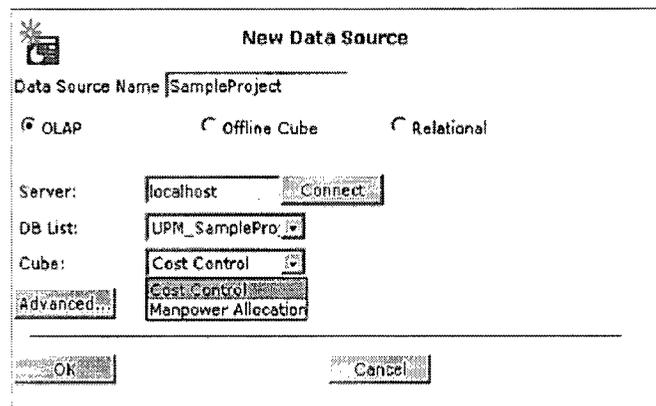


**Figure 7.1: Main UI of Microsoft BI Portal**

The next step after loading the main user interface was to create the new data source. By clicking the "Data Source" from the "Tools" menu, a dialog with a data grid of all

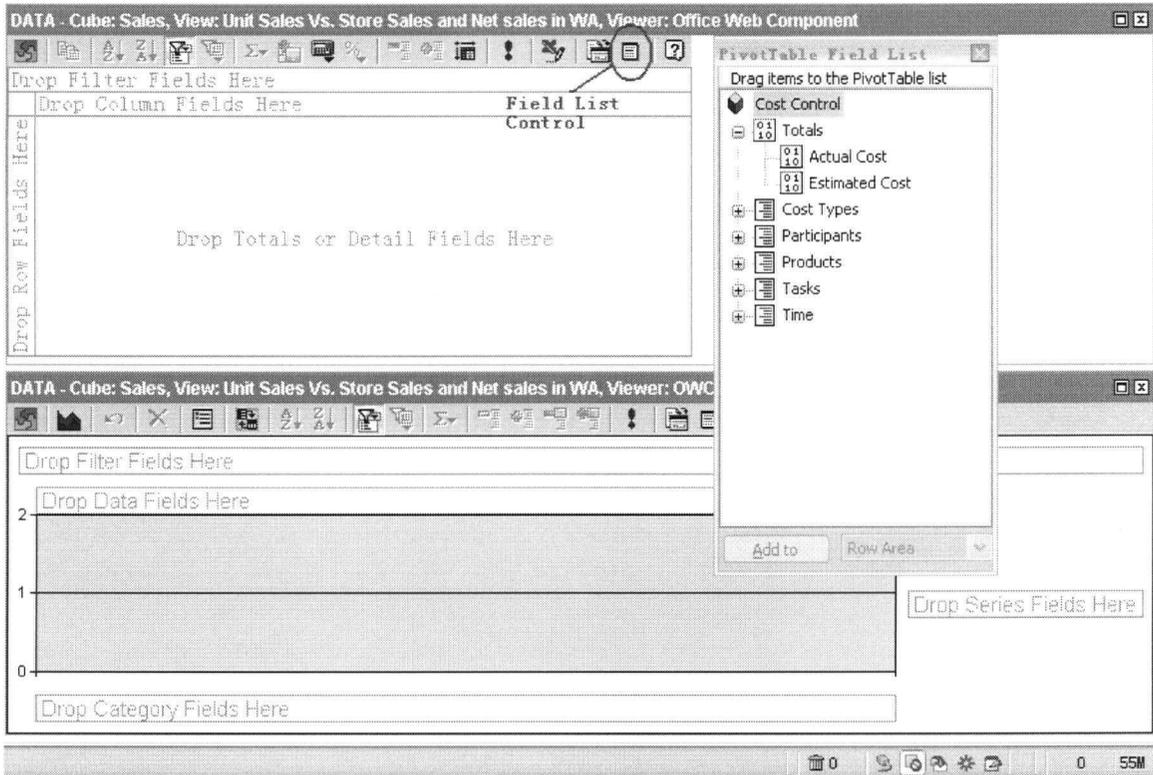
existing data sources displayed. For the purpose of testing, two new data sources were created—one for cost control and the other for manpower allocation.

Figure 7.2 is the screenshot of the dialog box displayed to add new data sources. The new data source for cost control testing was named “SampleProject”, and the one for manpower allocation was named “SampleProject1”. After adding “localhost” in the Server area and clicking the Connect button, the drop-down DB List included all databases on the localhost server. The previously established “UPM\_SampleProject” data warehouse was selected, and the cube list then provided two data cubes that had been built—a “Cost Control” cube and a “Manpower Allocation” cube.

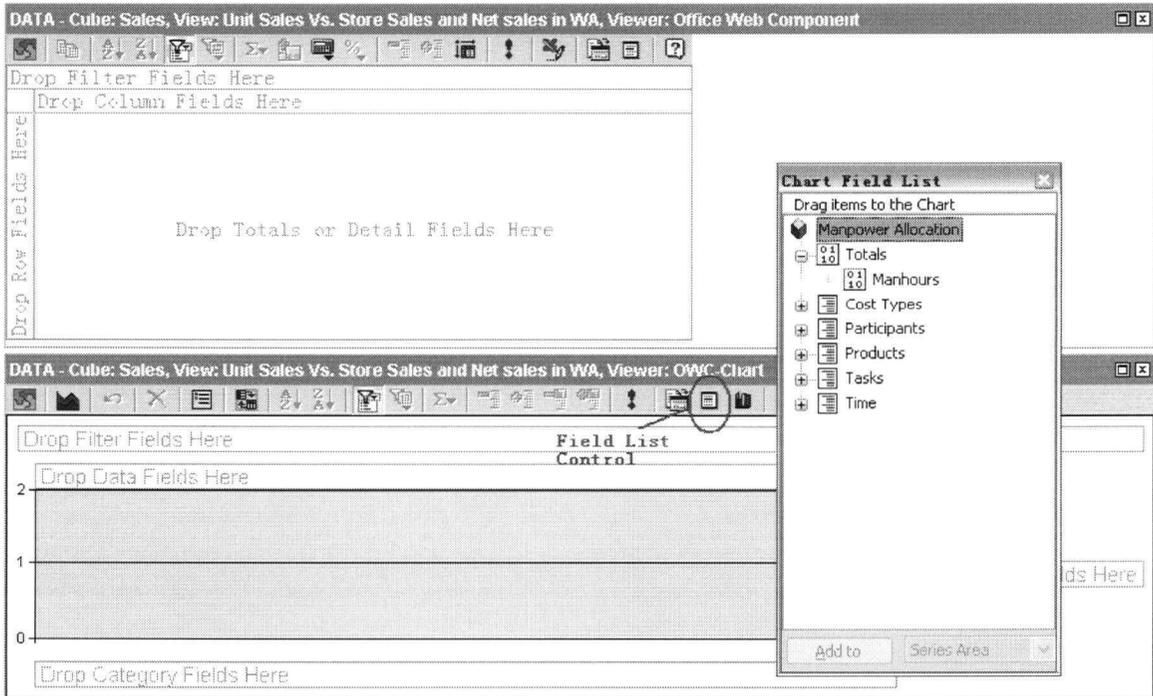


**Figure 7.2: Create New Data Sources for Cost Control and Manpower Allocation**

After creating the two new data sources, the next step was to create new data views for analysis. To do cost control testing, the “SampleProject” data source was connected. By clicking the Field List button from the OWC toolbar menu, the PivotTable/Chart field lists appeared with all measures and dimensions that had been defined (Figure 7.3 and Figure 7.4). To do manpower allocation testing, the above procedure was repeated, except for linking the application with the other data source “SampleProject1”.



**Figure 7.3: Link Data Source for Cost Control**



**Figure 7.4: Link Data Source for Manpower Allocation**

## **7.3. Testing & Evaluation – Project Cost Control**

### **7.3.1. Overview**

During the execution of a project, cost tracking and control is indispensable to project managers and other participants in the construction process. Project cost control serves the dual purpose of tracking financial transactions as well as giving project managers an indication of the progress and problems associated with the project progress.

The testing procedure on project cost control focuses on two types of cost records. The Estimated (Budgeted) Cost is derived from the detailed cost estimate prepared at the start of the project. The Actual Cost incurred in each category is derived from the financial record keeping accounts. By comparing those two cost records, the cost variance (CV) indicates a deviation of project cost.

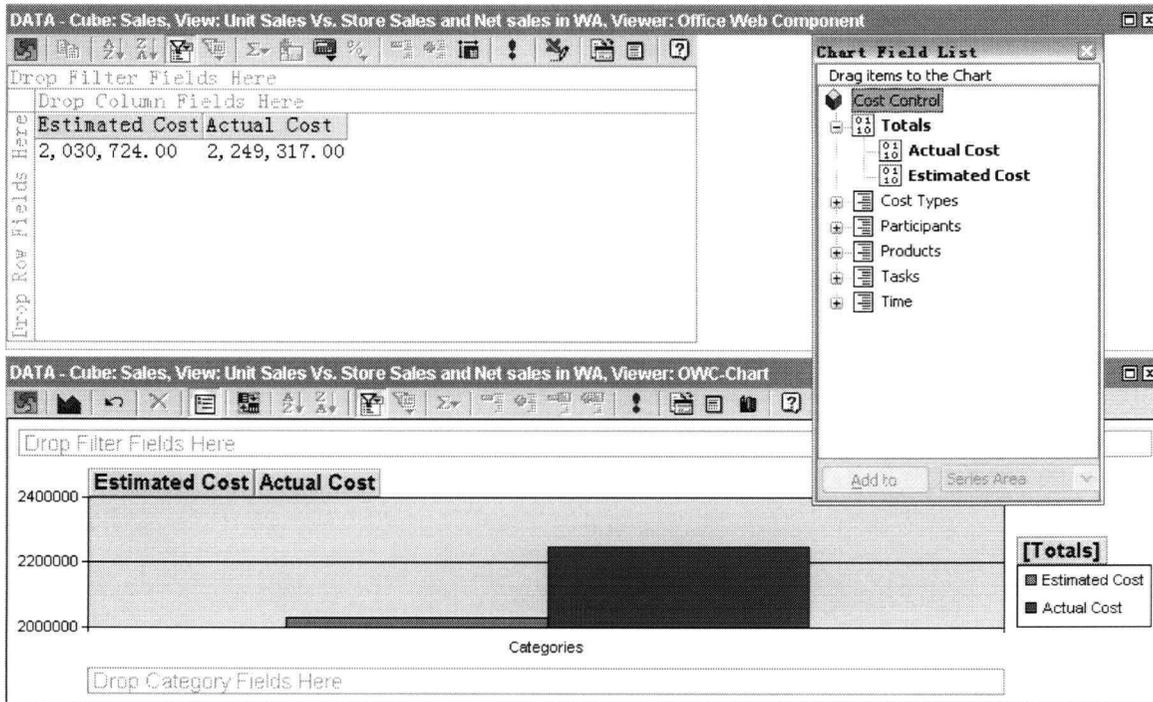
The following testing is organized by considering each of the major OLAP features, and within these, presenting several queries in the form of project management scenarios.

### **7.3.2. Cost Data Drill Through**

#### **7.3.2.1. Scenario 1 – High Level Cost Overview (Data Roll-up)**

The users may want quick answers to the following questions: “What is the total estimated (budget) cost and total actual cost of the foundation project?” “How is the total actual cost compared with the total estimated cost?”

To answer the above queries, the “Estimated Cost” and “Actual Cost” measures were first selected respectively from the “Cost Control” data cube field list in the main UI of the MS BI Portal. Then these two measures were dragged and dropped to the OWC area (Figure 7.5). As shown, the estimated cost of the whole project was \$2,030,724 while the actual total cost was \$2,249,317.

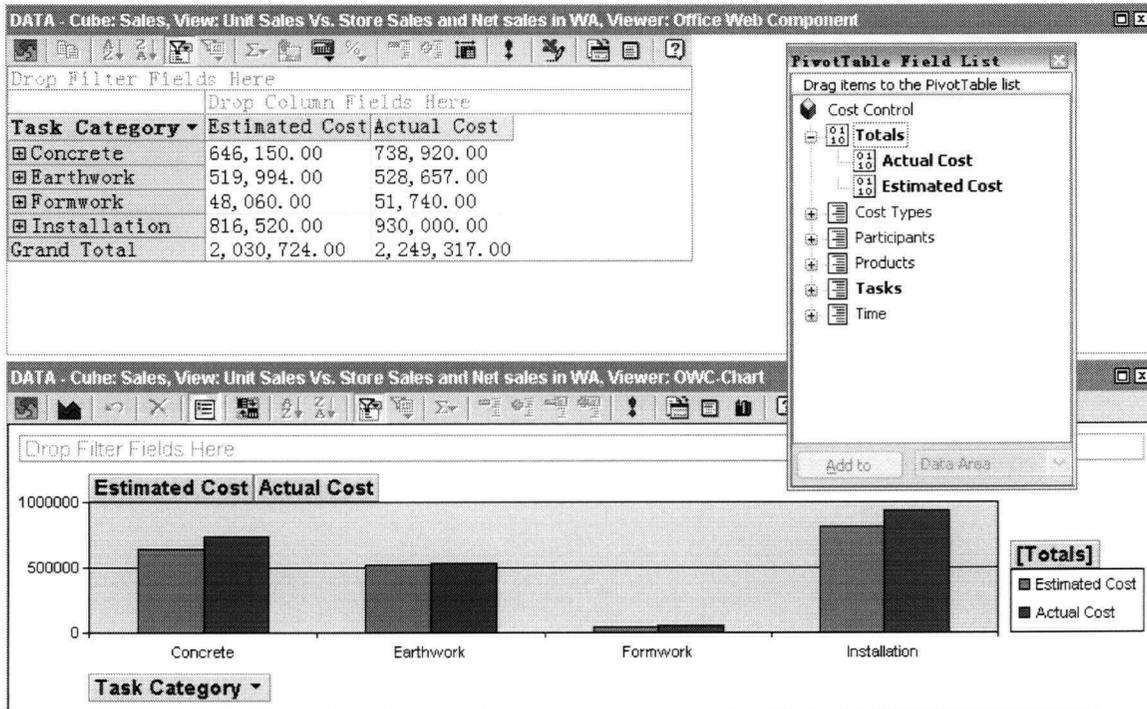


**Figure 7.5: Cost Control – Data Overview**

### 7.3.2.2. Scenario 2 – Low Level Cost Details (Data Drill-down)

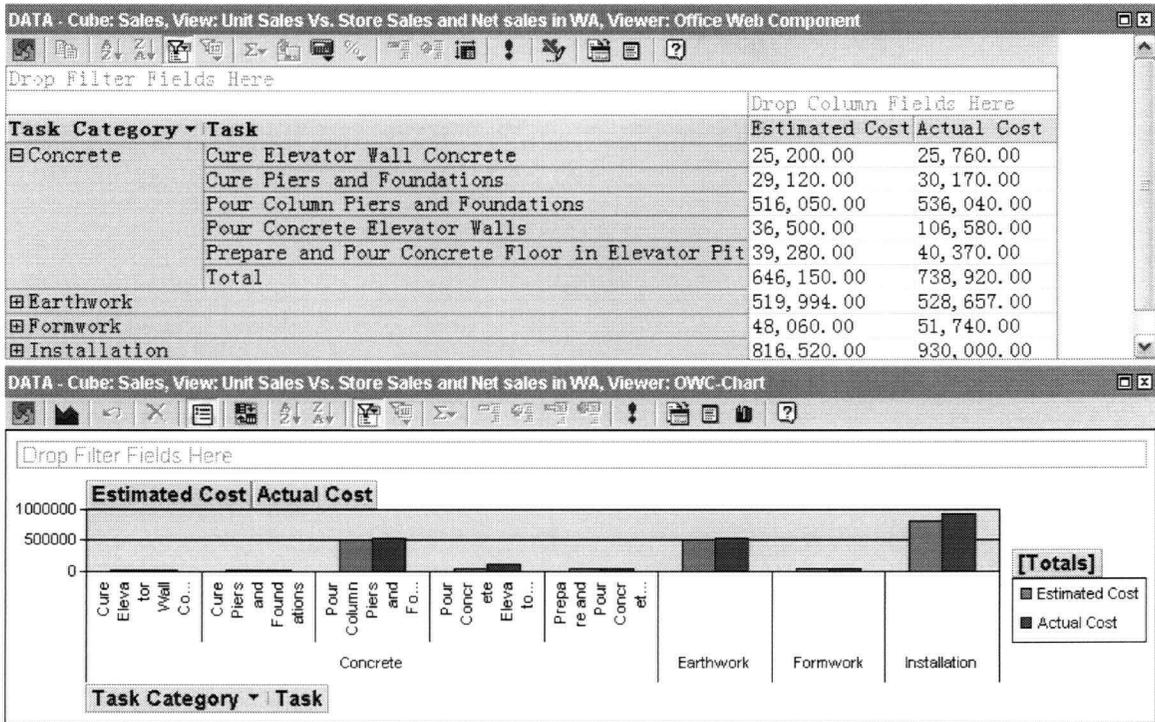
The users may be interested in cost details by tasks, and want quick answers to the following questions: “What are the estimated cost and actual cost of the concrete work?” “To explore more deeply, what is the estimated and actual cost of curing concrete for elevator walls?”

To answer the first question above, the “Estimated Cost” and “Actual Cost” measures from the “Cost Control” data cube were selected, dragged and dropped to the OWC area. Then the “Tasks” dimension was dragged to the “Row Field” on the left. That way the total estimated cost and actual cost were distributed as per task categories, which was the first level of the “Tasks” dimension (Figure 7.6). As shown, the estimated cost for concrete was \$646,150 and actual cost was \$738,920.

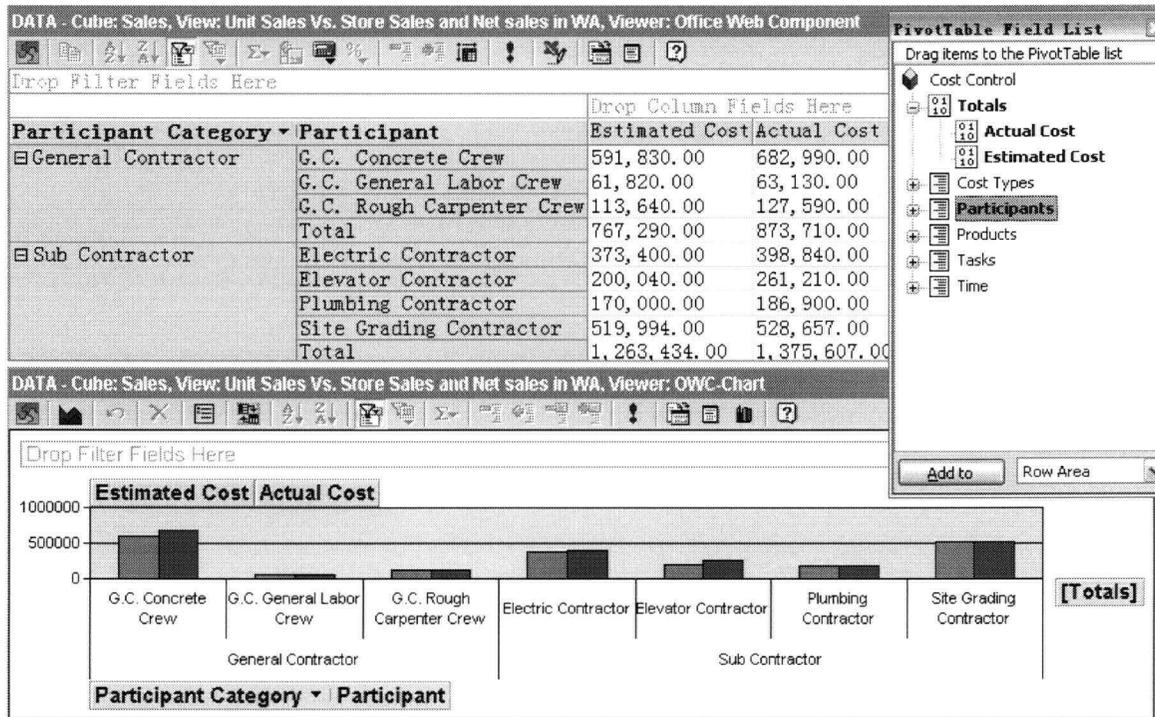


**Figure 7.6: Drill Down “Tasks” Dimension – Level one**

To answer the second question, the “Concrete” task category was expanded to list the lower level details underneath (Figure 7.7). As shown, the estimated and actual costs for “cure elevator wall concrete” were \$25,200 and \$25,760 respectively. The chart component (histogram) changed simultaneously when the cost data in PivotTable changed.

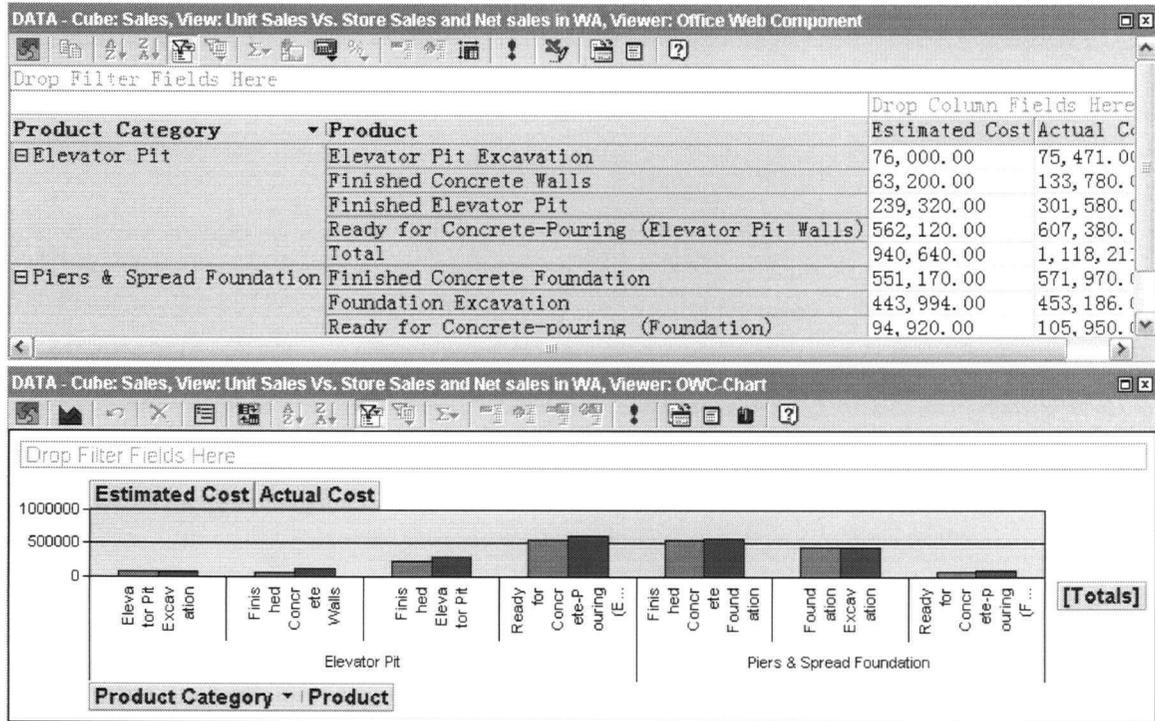


**Figure 7.7: Drill Down “Tasks” Dimension – Level two**



**Figure 7.8: Drill Down “Participant” Dimension**

To explore the cost information by participant or product, instead of by task, the specified dimension was dragged and dropped to the row field in the OWC area. Figure 7.8 is the screenshot of drilling cost information through the “Participant” dimension, and Figure 7.9 is the screenshot of drilling cost information through the “Product” dimension.



**Figure 7.9: Drill Down “Product” Dimension**

### 7.3.2.3. Comments on Cost Data Drill Through

From testing Scenario 1, the first impressive feature of OLAP-based cost control is its “drag & drop” capability. After the OLAP cube is defined and pre-processed in the Analysis Services, users can obtain feedback of their information queries very quickly by dragging and dropping data fields into the display areas. This allows cost information to be presented in a very dynamic and ad hoc manner. In addition, OLAP supports different measures in a same data cube—in this case the estimated cost and actual cost—to be presented and manipulated simultaneously. In the cost control domain, this helps project managers to track the cost variance between estimates and real occurrences at any level

of cost detail.

From the testing of Scenario 2, it can be concluded that, compared with a traditional paper-based cost list or a two dimensional electronic spreadsheet, the OLAP-based solution is more dynamic and fast for cost report rollup and drill-down. When going through a large quantity of cost data, users usually have to follow one or more paths from the top level to the bottom. The paths may either be time, task, participant, or product, etc. In a traditional two-dimensional spreadsheet, generally the cost data is organized and aggregated by only one path—mostly by products or tasks. It is very hard for users to switch from one path to another conveniently without re-aggregation and re-organization. For the OLAP-based solution, however, users can explore cost details in whatever path they want just by dragging and dropping the relevant dimensions to the OWC area. In addition, users can get query responses instantly, because of the pre-aggregation and pre-processing of data cubes. All possible query answers are already available before queries arise.

Moreover, the Chart component can work simultaneously with the PivotTable component. As tested here, users can get a graphical view of cost information at the same time when they read the cost numbers. This provides a very intuitive view of cost data that greatly helps the users in analyzing cost data and tracking cost distribution. The graphical presentation makes it easier for a quick grasp of the overall cost trend and distribution out of a large volume of cost data.

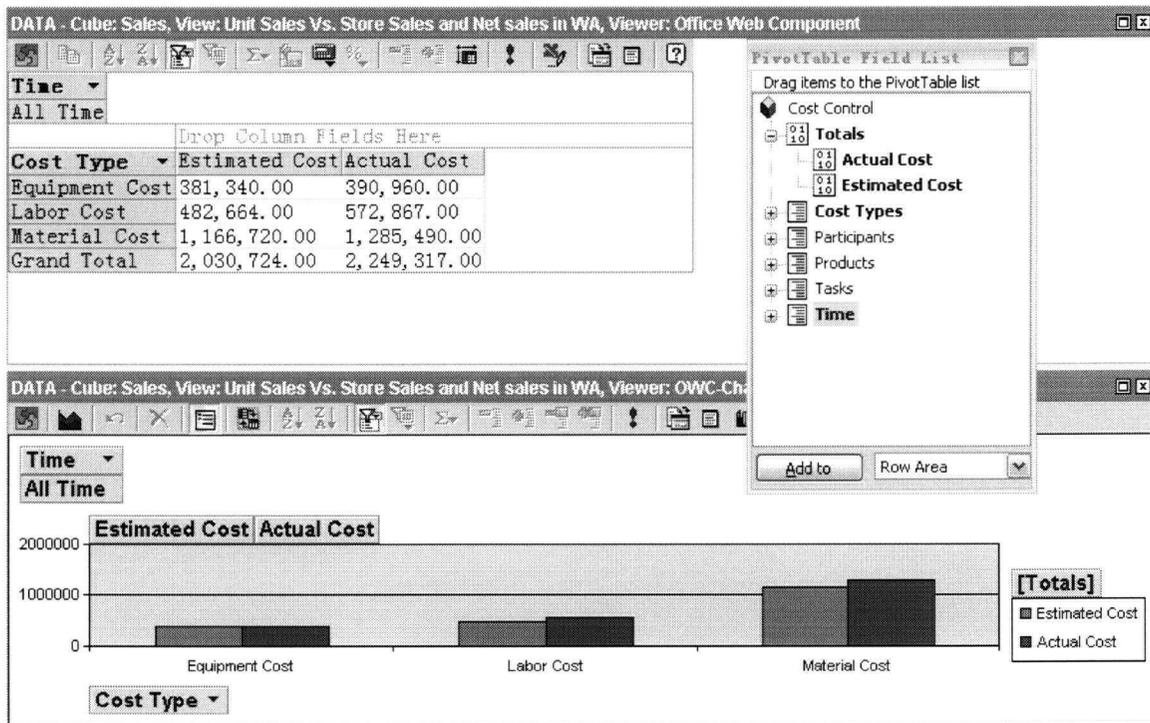
### **7.3.3. Cost Data Slice**

#### **7.3.3.1. Scenario 3 – Slice by Time:**

Sometimes users may want to investigate the cost information for a particular day. They may need that information to create daily cost reports, to do site space and layout planning, to arrange material supply and handling, or to adjust the project schedule. In these cases they may be especially interested in the following questions:

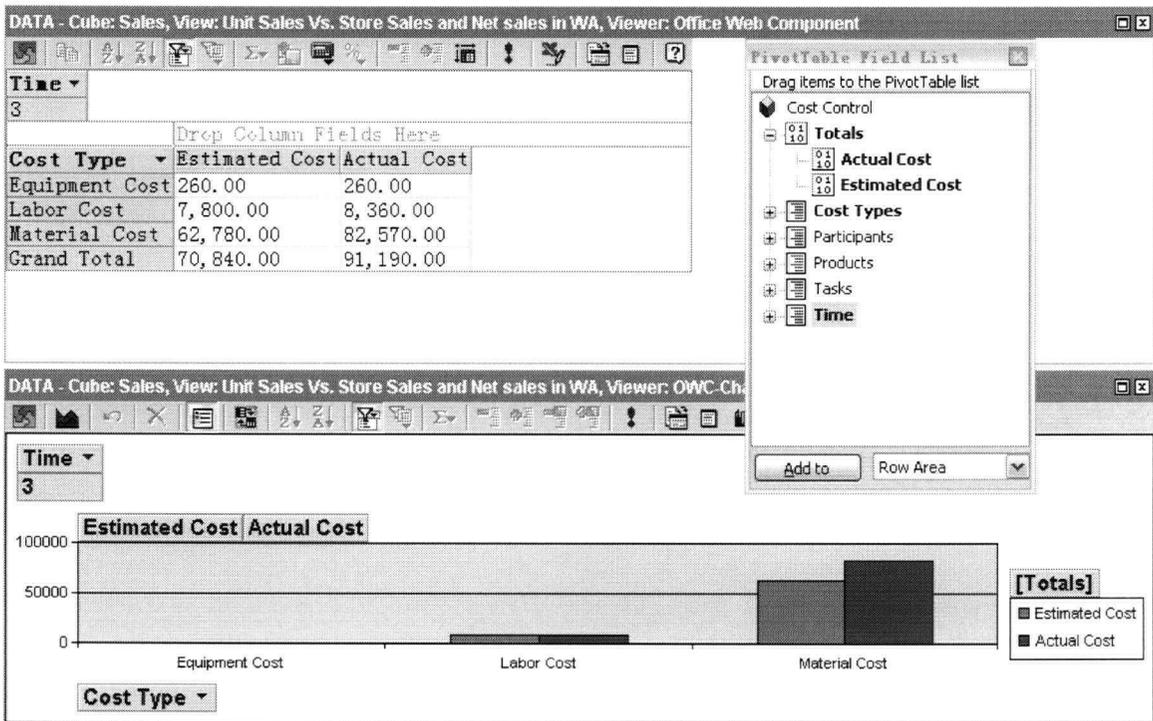
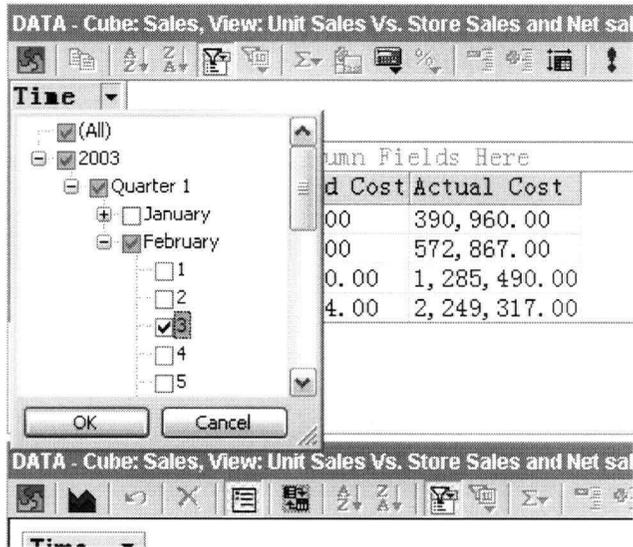
- 1) What is the estimated and actual equipment cost, labour cost and material cost respectively for a particular day, say Feb 3<sup>rd</sup>, 2003?
- 2) How many participants perform on that day? Who are they and who contributes most in terms of cost?

To answer the first question, the first step was to drag the “Cost Types” dimension to the left “Row Field”, and drag the “Time” dimension to the “Filter Field” on the top (Figure 7.10). The overall cost distribution by cost type was displayed for the whole project duration.



**Figure 7.10: Cost Info against Cost Types before “Slicing” by Time**

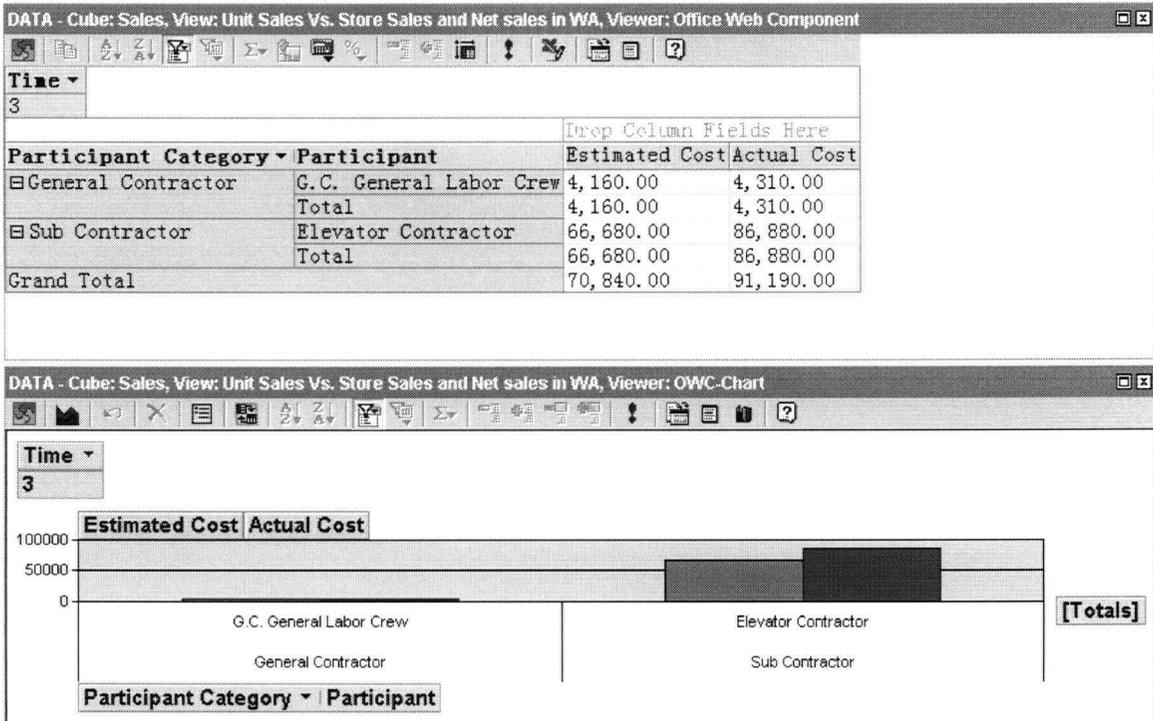
Then, the date of Feb 3<sup>rd</sup> was selected from the drop-down list of the “Time” dimension. All boxes were unchecked except for the one for Feb 3<sup>rd</sup> (Figure 7.11). As shown, the cost information for Feb 3<sup>rd</sup> was sliced out and still categorized by cost type. The requested equipment cost, labour cost and material cost for Feb 3<sup>rd</sup> were displayed.



**Figure 7.11: Cost Info against Cost Types after “Slicing” by Time**

To answer the second question, the “Participants” dimension was dragged to the row field so that it replaced the “Cost Types” dimension. Then it was expanded to the bottom level to get the answer (Figure 7.12). As displayed, there were two groups (participants) working on that day (February 3<sup>rd</sup>), which were the GC general labour crew and the Elevator sub-contractor. The estimated cost and actual cost for the GC general labour

crew were \$4160 and \$4310 respectively, while those costs for elevator sub-contractor were \$66,680 and \$86,880 respectively.



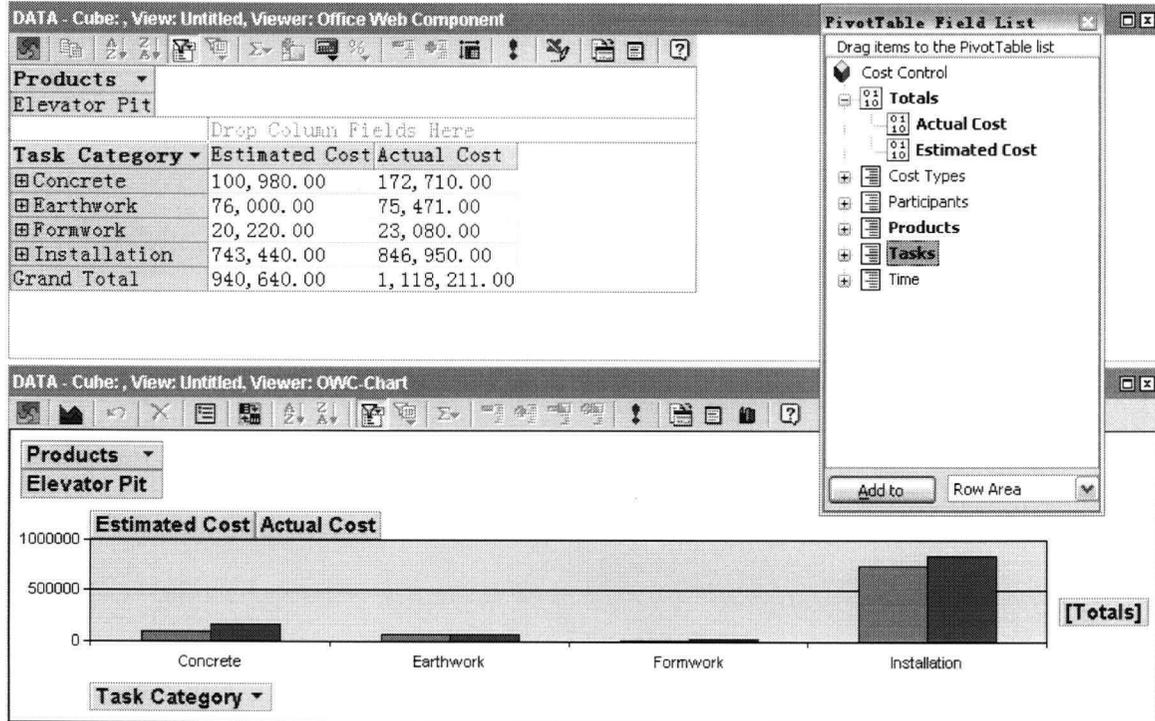
**Figure 7.12: Cost Info against Participants after slicing by “Time”**

### 7.3.3.2. Scenario 4 – Slice by Product:

More interestingly, project managers may prefer to track cost information by products—as long as the product model (dimension) has been properly established during OLAP cube design. For example, suppose a user would like to focus on and slice the data cube by the product of “Elevator Pit”. The similar process was repeated as in Scenario 3. The “Products” dimension was dragged to the filter field and “Elevator Pit” was selected to slice (Figure 7.13).

As shown in Figure 7.13, there were four task categories related to the “Elevator Pit” product—concrete, earthwork, formwork, and installation. If the entire task categories were expanded (Figure 7.14), the complete list of tasks contributed to accomplish the

“Elevator Pit” product was displayed, along with detailed cost values for each task.



**Figure 7.13: Cost Info against Tasks after slicing by “Products” – Overview**

Task Category	Task	Estimated Cost	Actual Cost
Concrete	Cure Elevator Wall Concrete	25,200.00	25,760.00
	Pour Concrete Elevator Walls	36,500.00	106,580.00
	Prepare and Pour Concrete Floor in Elevator Pit	39,280.00	40,370.00
	Total	100,980.00	172,710.00
Earthwork	Excavate Elevator Pit	76,000.00	75,471.00
	Total	76,000.00	75,471.00
Formwork	Form Elevator Pit Walls	18,720.00	21,640.00
	Strip Wall forms	1,500.00	1,440.00
	Total	20,220.00	23,080.00
Installation	Install Pneumatic Tube in Elevator Pit	200,040.00	261,210.00
	Rough-in Electric and Plumbing in Elevator	543,400.00	585,740.00
	Total	743,440.00	846,950.00
Grand Total		940,640.00	1,118,211.00

**Figure 7.14: Cost Info against Tasks after slicing by “Products” – Details**

### 7.3.3.3. Comments on Cost Data Slice

Because different project team players may have different focus of interest and may face different circumstances, it is not uncommon for a project manager or a cost engineer to track project cost from a specific and unique point of view. In the current project management process, however, the project cost report is always organized and distributed in a single format. It is very hard for a unique user-specified view of cost information to be conveniently accessed, extracted and thus represented for analysis.

As for the proposed OLAP-based reporting system, as tested in Scenario 3 and 4, users can easily get the exact piece of data slice that they need to fulfill their own analysis requirements. Technically, a data slice is a subset of a multi-dimensional array corresponding to a single value for one or more members of the dimensions not in the subset<sup>7-1</sup>. For example, if the member “Earthwork” is selected from the “Tasks” dimension as a slicing criterion, then the sub-cube of all the remaining dimensions consists of the slice that is specified to Earthwork task category. The data excluded from this slice would be any data associated with the non-selected members of the “Tasks” dimension, for example “Formwork”, “Concrete”, “Installation”, etc.

Besides the cost control scenarios tested, i.e. cost data slicing by time and by product, there are many other slicing ways that are also valuable. For example, some users may want to take a view of the labour cost exclusively from material and equipment. In that case, the users can slice the data cube by “cost type” and select labour cost as a criterion. For another example, it may also be very likely for some users to be especially interested in cost information that only relates to concrete work. In that case, the users can slice the data cube by task and select the concrete category to filter.

In summary, the OLAP-based solution provides a powerful and dynamic data slicing feature that is very useful for efficient project cost control. Unlike the traditional cost

---

<sup>7-1</sup> Definition from <http://services.eliteral.com/glossary/olap-and-olap-server-definitions.php>

reports, complex cost data can be sliced easily in the OLAP-based system to meet user's specific analysis needs.

### 7.3.4. Cost Data Slice & Dice

In the foregoing cost analysis scenarios, the cost data are organized through one dimension, although in scenario 3 and 4 an additional dimension is also involved to slice specific data out. In many cases, however, the users need to go more deeply into the details of data, which may need the cost values to be represented in a two-dimensional or even a multi-dimensional way.

#### 7.3.4.1. Scenario 5 – Two-dimensional Cost Report:

From a user's perspective, the term "slice" most often refers to a two-dimensional spreadsheet selected from the cube. The two-dimensional cost report is commonly used in the real-world industry.

To establish a traditional looking matrix that serves as a summary report of the daily costs in the construction industry, the "Time" dimension and "Cost Type" dimension were dragged respectively to the row field and column field (Figure 7.15).

		Equipment Cost		Labor Cost		Material Cost		Grand Total		
Month	Day	Estimated Cost	Actual Cost	Estimated Cost	Actual Cost	Estimated Cost	Actual Cost	Estimated Cost	Actual Cost	
January	6	42,210.00	42,280.00	18,090.00	17,639.00			60,300.00	59,919.00	
	7	44,600.00	44,600.00	19,134.00	20,632.00			63,734.00	65,232.00	
	8	36,400.00	36,900.00	37,150.00	37,180.00	114,300.00	114,900.00	187,850.00	188,980.00	
	9	36,400.00	36,400.00	37,150.00	39,500.00	114,300.00	124,800.00	187,850.00	200,700.00	
	10	40,600.00	42,000.00	38,950.00	40,080.00	114,300.00	124,800.00	193,850.00	206,880.00	
	13	42,700.00	42,700.00	39,850.00	43,290.00	114,300.00	128,000.00	196,850.00	213,990.00	
	14	40,900.00	40,900.00	21,620.00	22,390.00	820.00	820.00	63,340.00	64,110.00	
	15	37,380.00	39,000.00	19,880.00	21,380.00	820.00	820.00	58,080.00	61,200.00	
	16	23,030.00	25,000.00	18,210.00	19,754.00	1,800.00	1,800.00	43,040.00	46,554.00	
	17	19,600.00	19,600.00	16,740.00	18,712.00	1,800.00	1,800.00	38,140.00	40,112.00	
	20	1,600.00	1,580.00	12,580.00	77,800.00	27,780.00	32,980.00	41,960.00	112,360.00	
	21	500.00	480.00	14,730.00	16,100.00	12,100.00	12,980.00	27,330.00	29,560.00	
	22	500.00	480.00	10,250.00	11,600.00	11,120.00	12,800.00	21,870.00	24,880.00	
	23	500.00	480.00	10,250.00	11,600.00	11,120.00	12,800.00	21,870.00	24,880.00	
	24	500.00	480.00	10,250.00	11,600.00	11,120.00	11,850.00	21,870.00	23,930.00	
	27	2,300.00	2,470.00	22,290.00	23,900.00	82,220.00	73,100.00	106,810.00	99,470.00	
	28	2,300.00	2,970.00	22,290.00	24,100.00	82,220.00	82,400.00	106,810.00	109,470.00	
	29	2,300.00	3,470.00	22,290.00	27,000.00	82,220.00	87,400.00	106,810.00	117,870.00	
	30	1,800.00	2,890.00	21,790.00	21,240.00	81,120.00	96,000.00	104,710.00	120,130.00	
	31	1,800.00	2,880.00	25,490.00	25,600.00	142,600.00	161,900.00	169,890.00	190,380.00	
		Total	377,920.00	387,560.00	438,984.00	531,097.00	1,006,060.00	1,081,950.00	1,822,964.00	2,000,607.00
	February	3	260.00	260.00	7,800.00	8,360.00	62,780.00	82,570.00	70,840.00	91,190.00
		4	260.00	260.00	7,800.00	8,360.00	62,780.00	81,220.00	70,840.00	89,840.00

**Figure 7.15: Cost Report – “Time” vs. “Cost Type”**

Or, to establish a cost report that details the cost values (in terms of cost type) for each task, the “Tasks” dimension and “Cost Type” dimension were dragged to the row field and column field respectively (Figure 7.16).

Task Cat / Task		Cost Type				
		Equipment Cost		Labor Cost		Material Cost
		Estimated Cost	Actual Cost	Estimated Cost	Actual Cost	Estimated Cost
Concrete	Cure Elevator Wall Concrete	3,500.00	3,360.00	14,000.00	14,000.00	7,700.00
	Cure Piers and Foundations	1,820.00	1,820.00	18,200.00	17,360.00	9,100.00
	Four Column Piers and Foundations	9,000.00	13,240.00	101,450.00	108,600.00	405,600.00
	Four Concrete Elevator Walls	1,600.00	1,580.00	8,100.00	73,000.00	26,800.00
	Prepare and Pour Concrete Floor in Elevator Pit	1,600.00	1,580.00	9,080.00	6,890.00	26,600.00
	Total	17,520.00	21,580.00	150,830.00	219,850.00	477,800.00
Earthwork	Excavate Elevator Pit	53,200.00	52,200.00	22,800.00	23,271.00	
	Excavate Foundation	310,620.00	317,180.00	133,374.00	136,006.00	
	Total	363,820.00	369,380.00	156,174.00	159,277.00	
Formwork	Form Column Piers and Spread Foundations			17,920.00	18,980.00	3,920.00
	Form Elevator Pit Walls			15,440.00	18,360.00	3,280.00
	Strip Column Piers and Foundation Forms			6,000.00	5,760.00	
	Strip Wall forms			1,500.00	1,440.00	
	Total			40,860.00	44,540.00	7,200.00
Installat	Install Pneumatic Tube in Elevator Pit			15,600.00	17,560.00	184,440.00
	Rough-in Electric and Plumbing in Elevator			86,200.00	93,240.00	457,200.00
	Set Reinforcing and Anchor Bolts			33,000.00	38,400.00	40,080.00
	Total			134,800.00	149,200.00	681,720.00
Grand Total		381,340.00	390,960.00	482,664.00	572,867.00	1,166,720.00

**Figure 7.16: Cost Report – “Tasks” vs. “Cost Type”**

**7.3.4.2. Scenario 6 – Cost Data Slice & Dice (Multi-dimensional Report)**

In Scenario 4 (Figure 7.14), a data slice relating to the “Elevator Pit” product was sliced out, indicating all tasks (with cost information) involved to finish the elevator pit. Suppose a user needs more details on top of that. For example, he or she may need the equipment cost, labor cost and material cost for each task relating to the “Elevator Pit” product. The user may also want the daily cost information for this product.

To achieve the first request, the “Cost Type” dimension was dragged to the row field (or the column field) when the “Elevator Pit” product was still kept in the filter area. Then the detailed equipment cost, labor cost and material cost information was displayed (Figure 7.17).

To achieve the second request the “Time” dimension was dragged to the row field (or the column field) when the “Elevator Pit” product was kept in the filter area. Then the daily cost for each task relating to “Elevator Pit” was sliced out (Figure 7.18).

DATA - Cube: , View: Untitled, Viewer: Office Web Component				
Products ▾ Elevator Pit				
Task Category ▾ Task		Cost Type ▾	Estimated Cost	Actual Cost
Concrete	Cure Elevator Wall Concrete	Equipment Cost	3,500.00	3,360.00
		Labor Cost	14,000.00	14,000.00
		Material Cost	7,700.00	8,400.00
		Total	25,200.00	25,760.00
	Pour Concrete Elevator Walls	Equipment Cost	1,600.00	1,580.00
		Labor Cost	8,100.00	73,000.00
		Material Cost	26,800.00	32,000.00
		Total	36,500.00	106,580.00
	Prepare and Pour Concrete Floor in Elevator Pit	Equipment Cost	1,600.00	1,580.00
		Labor Cost	9,080.00	6,890.00
		Material Cost	28,600.00	31,900.00
		Total	39,280.00	40,370.00
Total			100,980.00	172,710.00
Earthwork	Excavate Elevator Pit	Equipment Cost	53,200.00	52,200.00
		Labor Cost	22,800.00	23,271.00
		Total	76,000.00	75,471.00
Total			76,000.00	75,471.00
Formwork	Form Elevator Pit Walls	Labor Cost	15,440.00	18,360.00
		Material Cost	3,280.00	3,280.00
		Total	18,720.00	21,640.00
	Strip Wall forms	Labor Cost	1,500.00	1,440.00
		Total	1,500.00	1,440.00
		Total	20,220.00	23,080.00
Installation			743,440.00	846,950.00

**Figure 7.17: Cost Report – “Product” vs. “Tasks” vs. “Cost Type”**

DATA - Cube: , View: Untitled, Viewer: Office Web Component					
Products ▾ Elevator Pit					
Task Category ▾ Task		Month ▾ Day	Estimated Cost	Actual Cost	
Concrete	Cure Elevator Wall Concrete	January	21	3,600.00	3,680.00
			22	3,600.00	3,680.00
			23	3,600.00	3,680.00
			24	3,600.00	3,680.00
			27	3,600.00	3,680.00
			28	3,600.00	3,680.00
			29	3,600.00	3,680.00
		Total	25,200.00	25,760.00	
	Total			25,200.00	25,760.00
	Pour Concrete Elevator Walls	January	20	36,500.00	106,580.00
			Total	36,500.00	106,580.00
		Total	36,500.00	106,580.00	
	Prepare and Pour Concrete Floor in Elevator Pit	February	5	39,280.00	40,370.00
			Total	39,280.00	40,370.00
		Total	39,280.00	40,370.00	
Total			100,980.00	172,710.00	
Earthwork	Excavate Elevator Pit	January	6	38,000.00	36,439.00
			7	38,000.00	39,032.00
		Total	76,000.00	75,471.00	
Total			76,000.00	75,471.00	
Formwork	Form Elevator Pit Walls	January	14	4,680.00	5,320.00
			15	4,680.00	5,320.00
			16	4,680.00	5,500.00

**Figure 7.18: Cost Report – “Product” vs. “Tasks” vs. “Time”**

Technically, the OLAP cubes can be sliced and diced from any dimension that is defined during the cube design. Because of that, the data report presentation can be dynamically manipulated in whatever way that the users expect. Sometimes the users may have to rely on some more sophisticated multi-dimensional cost reports for their cost analysis. For example, a user may write down the following specific requirements for his/her cost analysis:

- 1) “I only want to focus on cost data in January 2003.”
- 2) “I am only interested in the “Elevator Pit” product.”
- 3) “I want to list all tasks performed in January 2003 that are necessary to produce the elevator pit.”
- 4) “I would like to know who are doing those tasks (the participants).”
- 5) “I would like to understand the equipment cost, labor cost and material cost for each participant and for each task.”

Those requirements are actually user-specified criteria to do data slice and dice, by dealing with all the OLAP dimensions. To fulfil requirement 1, the “Time” dimension was dragged to the filter area and the “January 2003” member was selected. To fulfil requirement 2, the “Product” dimension was dragged to the filter area and the “Elevator Pit” member was selected. To fulfil requirement 3 and 4, the “Task” dimension and the “Participant” dimension were dragged to the row field. To fulfil requirement 5, the “Cost Type” dimension was dragged to the column field. Figure 7.19 is screenshot of the multi-dimensional cost report that complies with all of the user-specified requirements mentioned.

Task Cat	Task	Participant	Cost Type			
			Equipment Cost	Labor Cost	Material Cost	Grand Total
Concrete	Cure Elevator Wall Concrete	G.C. General Labor Crew	3,360.00	14,000.00	8,400.00	25,760.00
		Total	3,360.00	14,000.00	8,400.00	25,760.00
	Pour Concrete Elevator Walls	G.C. Concrete Crew	1,580.00	73,000.00	32,000.00	106,580.00
		Total	1,580.00	73,000.00	32,000.00	106,580.00
Total		4,940.00	87,000.00	40,400.00	132,340.00	
Earthwork	Excavate Elevator Pit	Site Grading Contractor	52,200.00	23,271.00		75,471.00
		Total	52,200.00	23,271.00		75,471.00
Total			52,200.00	23,271.00		75,471.00
Formwork	Form Elevator Pit Walls	G.C. Rough Carpenter Crew		18,360.00	3,280.00	21,640.00
		Total		18,360.00	3,280.00	21,640.00
	Strip Wall forms	G.C. General Labor Crew		1,440.00		1,440.00
		Total		1,440.00		1,440.00
Total			19,800.00	3,280.00	23,080.00	
Installation	Install Pneumatic Tube in Elevator Pit	Elevator Contractor		5,800.00	83,000.00	88,800.00
		Total		5,800.00	83,000.00	88,800.00
	Rough-in Electric and Plumbing in Elevator Pit	Electric Contractor		50,840.00	348,000.00	398,840.00
		Plumbing Contractor		42,400.00	144,500.00	186,900.00
Total			93,240.00	492,500.00	585,740.00	
Grand Total			57,140.00	229,111.00	619,180.00	905,431.00

**Figure 7.19: Multi-dimensional Cost Report with All Dimensions Involved**

### 7.3.4.3. Comments on Cost Data Slice & Dice

As an OLAP terminology, “slice & dice” refers to the user-initiated process of data navigating by calling for page displays interactively, through the specification of slices via rotations and drill down/up.<sup>7-2</sup>

To some extent, data slice & dice is the most attractive feature for an OLAP-based cost data presentation. As tested in Scenario 5 and 6, the proposed OLAP solution can help users conveniently create user-specified multi-dimensional cost reports that allow in-depth investigations of the cost data. As long as the dimension model is properly defined during the cube design, almost every cost-related query can be answered via the multi-dimensional cost analysis through data slicing and dicing. What the users need to do is to translate their specific query requirements to the corresponding dimensional manipulation criteria in order to have certain cost data sliced and diced. In addition, because the pre-calculated aggregation makes an OLAP cube potentially containing all the answers to possible queries, users can get responses almost instantly when they manipulate the data

<sup>7-2</sup> Definition from <http://services.eliteral.com/glossary/olap-and-olap-server-definitions.php>

cube.

The key task for cost management is to thoroughly understand the project's cost structure, before any analytic decisions can be made. In the current industry, most conventional cost reports are made mainly on two-dimensional spreadsheets. Generally those spreadsheets can only present the project's cost structure from one point of view in rigid way. The process of developing a traditional spreadsheet-based cost report is often labor-intensive and error-prone, especially for complicated project in terms of size and number of participants. On top of that, it will be very hard to trace, filter, rotate, or explore the cost presentation once this kind of cost report is completed.

The OLAP-based cost management system will be able to do nearly any task that the traditional tabulate-based cost reporting can do. But in contrast with conventional reporting, OLAP is much faster, more flexible and more thorough. Once the aggregation during cube processing is done, users can make the most of OLAP's "slice & dice" feature to develop their own cost structure presentations in an interactive manner. If the user is not satisfied with the current format produced, he or she can easily switch to another suitable report just by dragging and dropping. The maximum dimension number for an OLAP data cube is 126, and this is expected to be sufficient for even extremely complicated projects.

### **7.3.5. Advanced Cost Data Manipulation**

The OWC utilized in this OLAP-based cost control system can also support simple user-oriented query programming. For some advanced cost management requirements, the users may have to write SQL queries to get the required results.

#### **7.3.5.1. Scenario 7 – Calculation of BCWP and ACWP**

One important cost control technique is to develop the project budget and then track the actual spending against it. Generally the project manager prepares a time-phased budget,

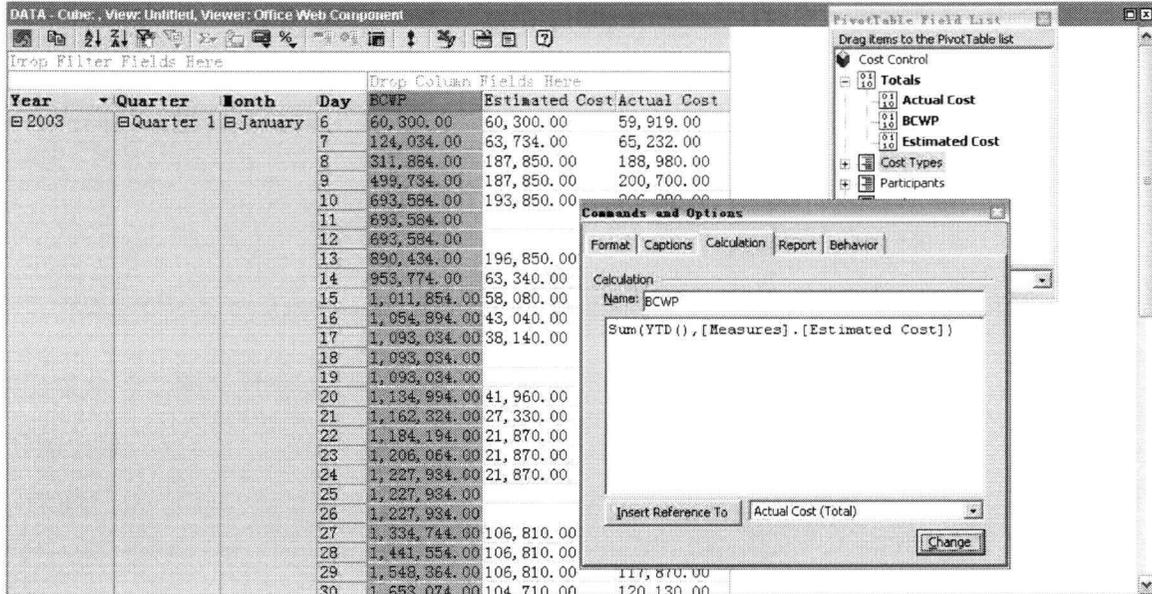
as done for the small sample project, breaking the overall budget into small intervals of days, weeks, or months. This can provide a budget plan for tracking actual costs against periodic budget targets.

In this scenario, suppose the project manager wants to calculate the accumulated cost data to come up with the budgeted cost of work performed (BCWP) and the actual cost of work performed (ACWP). After expanding the “Time” dimension to show detailed daily costs, the small calculator-like button was clicked and a new column was added to the OWC data field. In the “Commands and Options” pop-up box, the new calculation was named “BCWP”, and the following MDX expression was added in the query field (Figure 7.20):

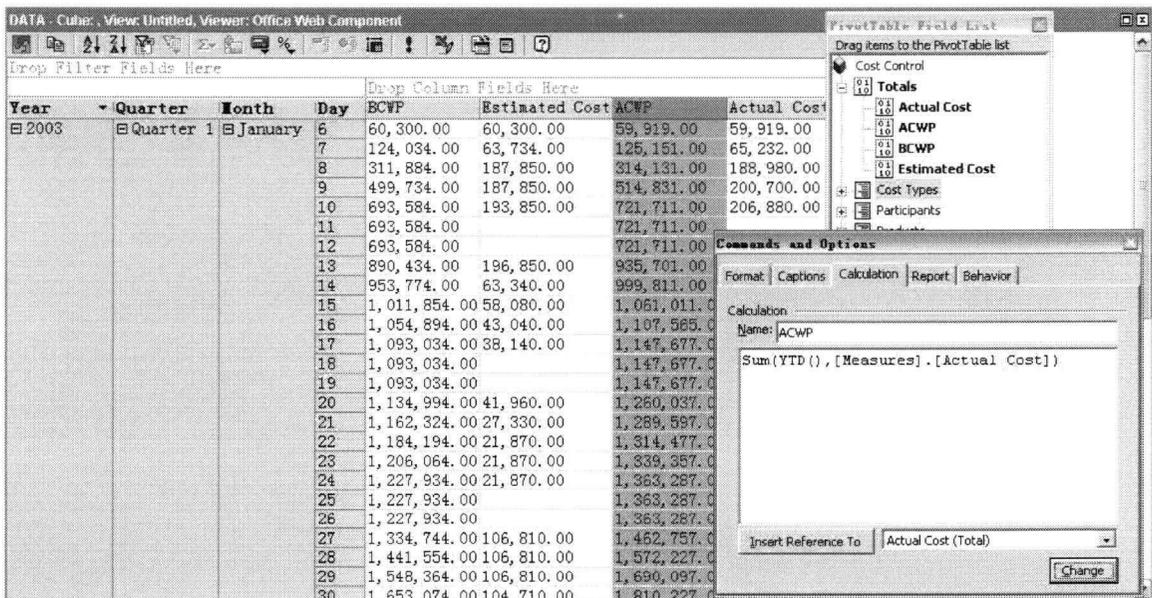
*Sum(YTD(),[Measures].[Estimated Cost])*

The YTD() function generates a set of days from the "Time" dimension up to the current member. The SUM function then adds the value of “Estimated Cost” for each day, and reports that total in the cell for that day. To calculate the ACWP values, the same procedure was followed, and the following MDX expression was used (Figure 7.21):

*Sum(YTD(),[Measures].[Actual Cost])*



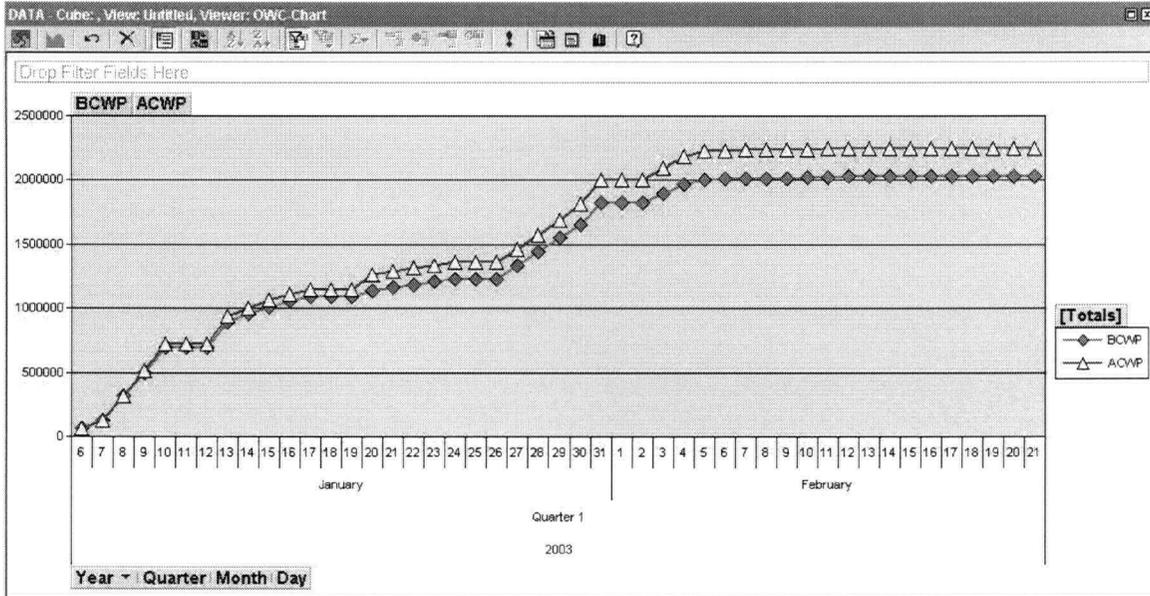
**Figure 7.20: Add New Calculation for BCWP**



**Figure 7.21: Add New Calculation for ACWP**

Because it was assumed that the sample project was on schedule, the budgeted cost of work scheduled (BCWS) was equal to the budgeted cost of work performed (BCWP). In that case, there was no schedule variance for the sample project. The cost variance can be easily calculated in the OWC area by a subtracting of the above two MDX expressions. The OWC Chart component displayed the cumulative budget cost and actual cost, both of

which usually appear in rough “S” curves (Figure 7.22). As shown, the ACWP curve was somewhat above the BCWP curve, which means the sample project was a little bit over budget.



**Figure 7.22: “S” Curve of BCWP and ACWP**

### 7.3.5.2. Comments on Advanced Cost Manipulation

As tested in scenario 7, the OLAP-based cost control system can successfully calculate accumulated cost, which is useful for cost track and control. Of course it may be easier for this to be accomplished manually in a traditional Excel spreadsheet, but the point is that for OLAP-based cost control system, further user-specified calculation is possible on top of all the aggregations and pre-calculations during OLAP cube processing.

Unfortunately, to do further programming after OLAP cubes have been defined and processed, users are required to have at least some basic knowledge of Microsoft's multi-dimensional query language – Multidimensional Expressions (MDX). They must also understand the multidimensional database structure of the data warehouse, as well as the concepts of “dimension”, “level”, “member”, “measure” and so on. This may be difficult for end users.

## **7.4. Manpower Allocation**

### **7.4.1. Overview**

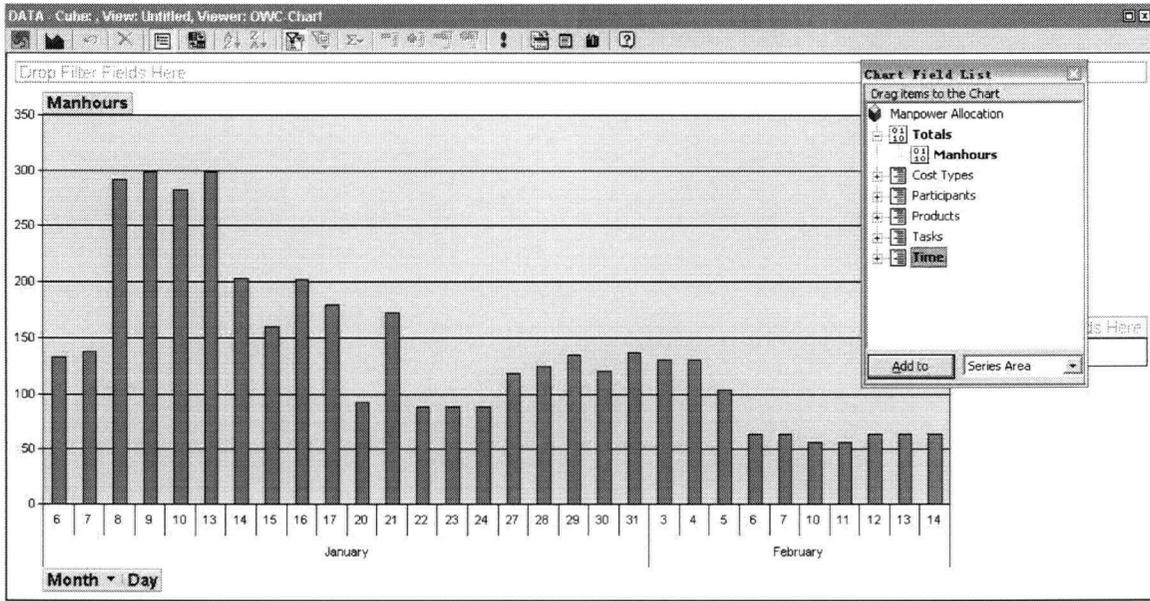
Another important requirement for project control is the resource loading of activities in the schedule. Forecasted activity duration may not have any validity before its resource loading has been conducted. This is because the formula that calculates the activity duration requires the application of the productive resource (e.g. manpower or equipment) against its projected productivity. How the resources are allocated over the project life cycle is a very important view for overall project control.

### **7.4.2. Scenario 8 - Manpower Allocation over Time**

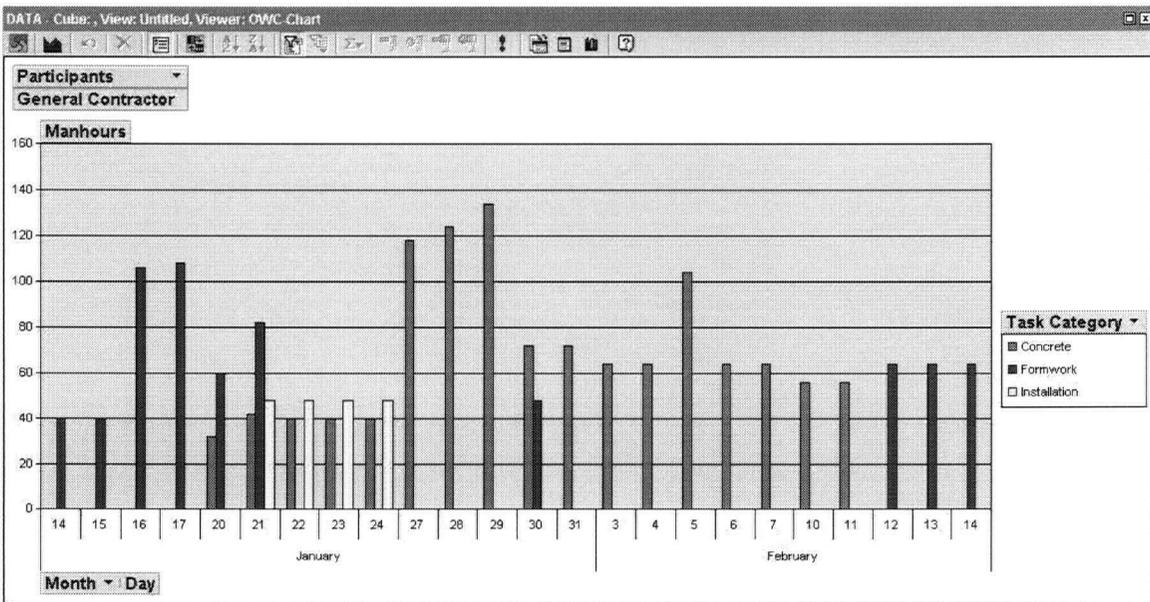
For many construction projects, the manpower may be the largest expense of all. The manpower distribution charts are useful for resource tracking and site planning for a construction project. Typically, a manpower allocation chart is depicted on a graph with one side for man-hours and the other side for the dates on which the labor is provided. To display this in the OLAP-based prototype, the “Time” dimension was dragged to the bottom category field (Figure 7.23).

With the help of OLAP’s “slice and dice” feature, the typical manpower distribution chart can be further manipulated toward a more user-oriented view. For example, suppose the project manager is interested in the manpower loading information for the general contractor only. He or she can slice the data cube by “General Contractor” from the “Participant” dimension. The user may also want to combine the “Time” dimension with the “Tasks” dimension, so that the detailed manpower consumption for each task category will be displayed.

To fulfil the user-specified manpower allocation view, the “Participant” dimension was dragged to the filter field, and the “General Contractor” member was selected. Then the “Task” dimension was dragged to the series field (Figure 7.24).



**Figure 7.23: Manpower Allocation over Time (Typical View)**

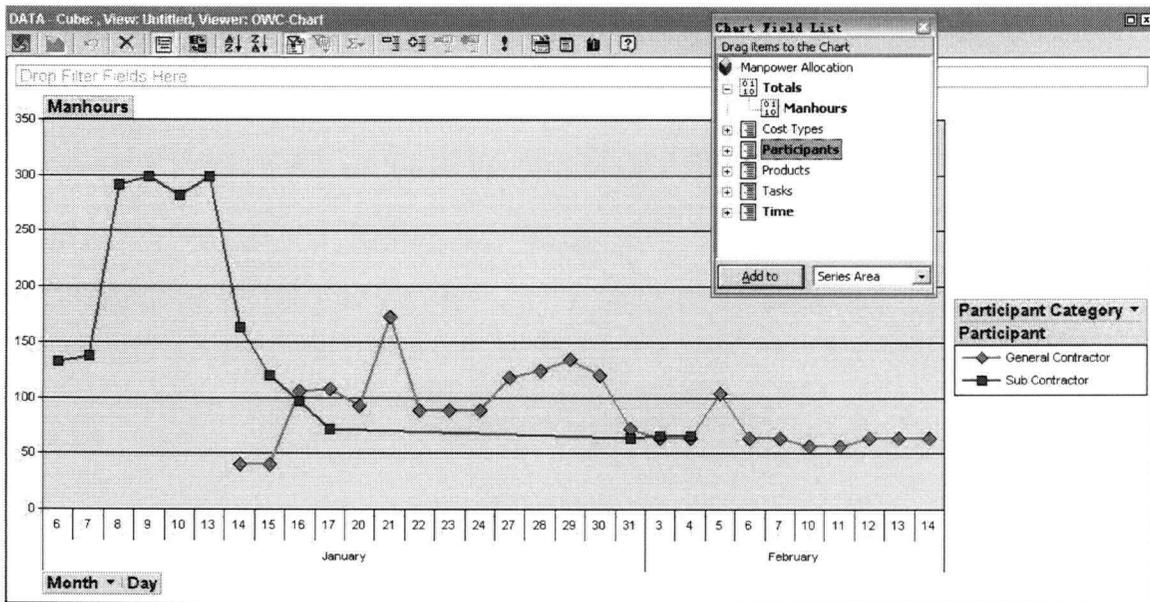


**Figure 7.24: Manpower Allocation over Time (User-specified View)**

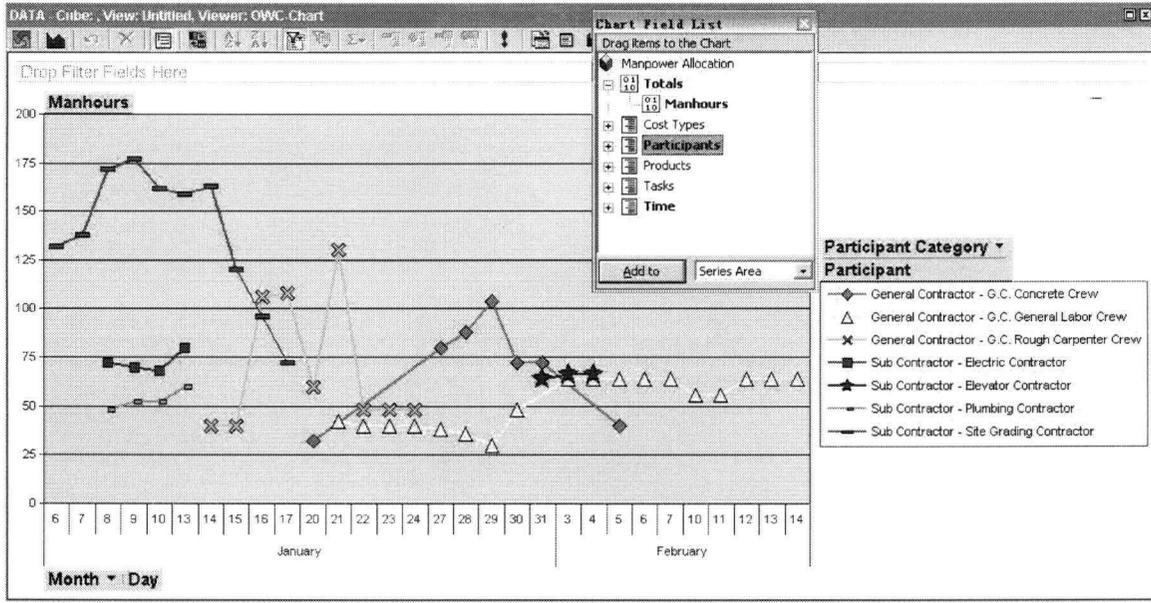
In the construction industry, every project has more than one participant involved, regardless of the size and type of the project. In addition to the overall manpower distribution over time, it is useful to understand the manpower allocation for each participant. To achieve this, the OWC chart type was first changed from “Histogram” to

“Line with Mark”. Then the “Participant” dimension was dragged to the series field. Figure 7.25 is the screenshot after the manipulation, which clearly shows the manpower loading trend for general contractor and subcontractors respectively.

If the user wishes to take further advantage of OLAP’s drill-down feature, he or she can expand the “General Contractor” and “Sub-contractors” categories, so that more detailed manpower distribution graph within each category will be displayed. Figure 7.26 shows the manpower allocation of concrete crew, general labor crew and rough carpenter crew under the General Contractor category, as well as the manpower allocation of electric contractor, elevator contractor, plumbing contractor and site grading contractor under the Sub-Contractor category.



**Figure 7.25: Manpower Allocation of Each Participant Category**



**Figure 7.26: Manpower Allocation of Each Participant**

### 7.4.3. Comments on OLAP-based Manpower Allocation

The OLAP-based project management system can run a variety of manpower loading reports in a flexible way. The OWC chart component supports most OLAP features such as drill through, data slice, or rotation. As tested in Scenario 7, users can slice data cube in order to create specific manpower distribution charts, or drill through a certain dimension for manpower details.

Under common circumstances in the construction management process, the manpower is generally distributed over time. However, if a user wants to show specific manpower distribution over task or over participant, he or she can achieve this in a very simple way, because both the “task” and the “participant” have been defined as dimensions of the manpower data cube. Theoretically, users can distribute manpower against any dimension they select as long as the final result chart is helpful.

## **CHAPTER 8: CONCLUSION AND FUTURE WORK TRENDS**

### **8.1. Introduction**

This chapter recapitulates the thesis research on an OLAP-based PM prototype for cost control and manpower loading analysis. The Pros and cons of applying the OLAP technology to project management are summarized. Future research work on the OLAP-based PM system is discussed.

### **8.2. OLAP-based PM System versus Conventional PM Systems**

Among the currently used project management systems in the industry, most of them do not support OLAP-related multidimensional data analysis. Some PM systems do support PivotTable presentation, such as the Microsoft Project Server 2003. Even so, however, there are still significant differences between the MS Project Server and the proposed OLAP-based PM system.

First, the data storage and handling method is totally different. The MS Project Server, as most other PM systems, uses a conventional relational database to store project data. The proposed OLAP-based PM system, on the contrary, unloads project data from an OLTP database and reorganizes it to a data warehouse with a multi-dimensional structure. After that, data cubes are defined and extracted from the stationary project data, using the OLAP technology.

Secondly, the data presentation method is different. The MS Project Server is primarily for scheduling, which is largely activity-oriented. The project data is mostly delivered in a two-dimensional way. Even its PivotTable presentation is generally established after the schedule is complete. The OLAP-based PM system, however, represents project data in a multi-dimensional manner. There are many possible project views (dimensions) to be represented, and the “tasks” dimension is only one of them.

Moreover, although the MS Project supports PivotTable presentation, it lacks clear definitions of data cubes, dimensions, dimension levels, and measures. Generally, it is very hard for end-users to add a new dimension in the MS Project, such as the “Product” dimension that is not included by default but has an important role for project management. Even for those dimensions that are initially included in the MS Project, it is difficult to further define their hierarchies—for example, the hierarchy of the “Participant” dimension—that are essential for data analysis. The OLAP-based system, on the contrary, allows any new dimension to be added, with any possible hierarchy definition, as long as the dimension is meaningful for the multi-dimensional data analysis.

### **8.3. Thesis Conclusion**

As a pilot prototype, this thesis research does not attempt to make the Web-based, OLAP-integrated PM system well-rounded with all possible functionalities that might be of value to end users. Rather, the main purpose is to do a feasibility study to test the idea of applying the fast growing OLAP technology to facilitate project management in the construction industry. From the process of developing the data warehouse, designing the OLAP cube, and testing and evaluating the system, we can see that OLAP technology does offer potential improvements to project management, although a huge volume of future work is needed.

The OLAP-based PM System benefits from various points:

- *Information versus Data:* To better control projects, the project managers or any other end-users need both information and data. Compared with the traditional solutions, the OLAP-based PM system provides the end user with more information rather than just data. OLAP makes it easy for users to identify trends or patterns behind the data very quickly, without the need for them to search through masses of “raw” data. The OLAP-based PM system is more analysis-

driven and information-rich, which is suitable for answering project control questions such as “How is the actual costs compared with the estimated costs of the concrete work for general contractor spent in January?”

- *Multi-dimensional Visualization:* In contrast with traditional cost control or manpower allocation reports, the OLAP-based PM system can produce multi-dimensional visualization of project data (costs, manpower, etc.). Since external information related to selected project data is structured in a multi-dimensional model, it decreases time spent explaining and understanding various aspects of the project’s status. With the help of OLAP-based chart tools—such as the OWC chart component—creative graphical representations of data can be developed, which help reduce the ambiguity of status and interrelationships among various aspects of projects. Because of this multi-view representation ability, the OLAP technology may be regarded as a possible way to move towards a well-rounded UPM (unified approach to project management) solution.
- *Fast Speed:* The fast response speed is one of the most attractive features for the OLAP-based PM system. A large construction project generally produces a huge volume of project data for analysis. With the help of the OLAP technology, query responses can be obtained in seconds regardless how complicated the data structure is.
- *Flexible Performance:* Because the volume of project data is always numerous, it is generally difficult for users to manipulate project data and develop ad hoc data presentation in a user-specified way. The OLAP-based PM system, however, provides a flexible user interface that helps produce ad hoc data presentation in a dynamic and intuitive manner. With OLAP’s slice and dice feature, users can produce any data presentation they want, often using only dragging and dropping.
- *User-specified Implementation:* OLAP-based PM system can be implemented

individually as per unique requirements of particular users. For different projects, the OLAP cubes may be designed in distinct ways so that the final data analysis can be performed through different data representations. Even if the OLAP cubes have already been set and processed, the end users still have the chance to create new fields with new data by writing MDX expressions.

The OLAP-based approach also has some shortcomings, and difficulties to extend its usability to a comprehensive PM system. The main issues include:

- *PM Application Limitations:* Not all project management application areas fit well within the OLAP-based PM framework, largely because of the data type requirement for OLAP cubes to process and manipulate. The project data that can be defined as the “measures” in OLAP cubes must be numeric, addable, and with consistent units. Besides cost control and manpower allocation, as tested in chapter 7, other PM areas that meet the requirement and can take advantages of the OLAP technology may include material handling, or equipment tracking for large-sized projects. Other PM areas, such as time management, or risk management, may be very hard to fit, because the produced project data does not meet the data type requirement of the OLAP technology.
- *Data Set Requirement:* An important requisite of the approach is the high-quality, fully interlinked data sets, which might be difficult to collect and organize into the data warehouse with a multi-dimensional data structure. For the proposed prototype, as stated in chapter 4, it mainly focused on the design of the data warehouse and the OLAP cubes, assuming the project data has been previously unloaded from an OLTP database. For a well-rounded OLAP-based system, however, the data structure of the OLTP database, as well as the data transmission services to offload project data, must be established to ensure the practicability of the OLAP-based multidimensional data representation. That process may be very

complicated, because for real projects, the produced project data may typically be very difficult to collect and organize in the necessary form.

Considering the main benefits and deficiencies of the proposed OLAP-based PM system, future research work to extend its applicability is discussed in the next section.

#### **8.4. Future Work**

Although this prototype may be regarded as the first attempt of integrating the OLAP technology for an UPM platform [Froese, 2], there is still a long way to go for an all-inclusive OLAP-integrated IT platform (i.e. a “Project Intelligence” solution) for project management in the construction industry. A great deal of work needs to be done in the future, which may include the following:

- *OLTP Design and Connection:* The project management practice produces a large volume of project data that keeps changing every day. One critical focus is on how to efficiently add, delete, change, and store those real-time operational data in OLTP databases, as well as how to offload data to the data warehouse through ETL services (Extract, Transform, and Load). The structure of the OLTP database must be carefully designed to ensure a clear representation of the project status. Because the daily operation data are always from different parts of different sites, it is critical to develop a friendly web-based UI for user’s easy manipulation.
- *Multidimensional Model and OLAP Design:* In the proposed prototype, a star schema was applied for the structure of the data warehouse. “Time”, “Task”, “Participants”, “Products” and “Cost Type” were defined as the OLAP dimensions. Costs and manpower were defined as the OLAP measures. However, it is still a critical focus to refine the design of the multidimensional structure of the data warehouse and the OLAP cubes. This includes the design of the data warehouse structure model, as well as the OLAP dimensions, levels and measures. The data

warehouse structure may be “star” or “snow-flake”; other possible OLAP dimensions may include “Purchase (Order)”, “Contracts”, or “Quality”; other possible OLAP measures may be like “material”, or “equipment”.

- *Updating Data Cubes:* Project information such as tasks or participants may change at any time, which means the structure of the pre-processed OLAP cubes, may need to be changed or updated frequently. Because of that, an important feature of the OLAP-based system is the possibility and convenience for the end users to update the cube structure whenever it is needed. Otherwise the OLAP-based PM system can only be used as a post-construction analysis or reporting tool rather than a real-time analysis tool to track and control the ongoing projects.
- *Front-end UI Design:* The front-end UI to represent project cube data should be particularly suitable for controlling construction projects. A lot of work needs to be focused on this area to develop dynamic graphical presentation for OLAP data yet in a common manner applicable to construction industry, such as Gantt charts or CPM diagrams.

## **BIBLIOGRAPHY**

- [Hendrickson, 1] Chris Hendrickson, Tung Au, "Project Management for Construction", Chapter 1 "The Owner's perspective", 2<sup>nd</sup> Edition 2003 for WWW Publication; 1<sup>st</sup> Edition 1989 printed by Prentice Hall
- [Froese, 2] Thomas Froese, Sheryl Staub-French, "A Unified Approach to Project Management", 4th joint symposium on information technology in civil engineering, November 2003, Nashville, USA.
- [Pendse, 3] Nigel Pendse, Richard Creeth, "OLAP Report Homepage", Web page at <http://www.olapreport.com/> (accessed July 2004)
- [Microsoft, 4] Microsoft MSDN Library, "Creating and Using Data Warehouses Overview",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/createdw/createdw\\_67qr.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/createdw/createdw_67qr.asp) (accessed June 2004)
- [Microsoft, 5] Microsoft MSDN Library, "Data Warehousing and OLAP",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agaboutolap\\_9i5s.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agaboutolap_9i5s.asp) (accessed June 2004)
- [Microsoft, 6] Microsoft MSDN Library, "Object Architecture",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agimplementolap\\_6a7i.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agimplementolap_6a7i.asp) (accessed July 2004)
- [Microsoft, 7] Microsoft MSDN Library, "Introduction to Dimensions",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agdimensions\\_30mr.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agdimensions_30mr.asp) (accessed July 2004)
- [Microsoft, 8] Microsoft MSDN Library, "Frequently Asked Questions about OLAP and Microsoft Analysis Services 2000",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmda/html/odc\\_olapfaq.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmda/html/odc_olapfaq.asp) (accessed July 2004)
- [Microsoft, 9] Microsoft MSDN Library, "Levels and Members",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/aglevels\\_7w4z.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/aglevels_7w4z.asp) (accessed July 2004)  
This webpage from Microsoft provides overview and definitions as an introduction of Microsoft SQL Server Analysis Services.
- [Microsoft, 10] Microsoft MSDN Library, "Aggregations",  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agaggregations\\_3tyr.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agaggregations_3tyr.asp) (accessed July 2004)

- [Microsoft, 11] Microsoft MSDN Library, "Key Concepts in MDX", <http://www.awprofessional.com/articles/article.asp?p=29418&seqNum=3> (accessed August 2004)
- [Pearson, 12] William E. Pearson, III, "Introduction to MDX: Basic Concepts and Navigation", Web Publication, September 2002, <http://www.awprofessional.com/articles/article.asp?p=29418&seqNum=3> (accessed August 2004)
- [Microsoft, 13] Microsoft MSDN Library, "Overview of PivotTable Services", [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/pt\\_intro\\_49wl.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/pt_intro_49wl.asp) (accessed August 2004)
- [Browning, 14] Dave Browning, Joy Mundy, "Data Warehouse Design Considerations", Chapter 17, SQL Server 2000 Resource Kit, Microsoft Corporation
- [Chaudhuri, 15] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology", SIGMOD Record, Vol. 26, No. 1, 1997, pp. 65-74.
- [Microsoft, 16] Microsoft MSDN Library, "Microsoft Office XP Developer – Web Components and Controls", (accessed August 2004) <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/modcore/html/deoriwebcomponetscontrols.asp>
- [Aouad, 17] G. Aouad, M. Betts, P. Brandon, F. Brown, T. Child, G. Cooper, S. Ford, J. Kirkham, R. Oxman, M. Sarshar, and B. Young, "Integration of Construction Information (ICON): Final Report - Integrated Databases for the design, Procurement and Management of Construction (University of Salford)", Salford, 1994
- [Rezgui, 18] Y. Rezgui, A. Brown, G. Cooper, G. Aouad, J. Kirkham and P. Brandon, "An Integrated Framework for Evolving Construction Models", The International Journal of Construction Information Technology, (4)1, 47-60 (1996)
- [Faraji, 19] I. Faraj, M. Alshawi, G. Aouad, T. Child and J. Underwood, "An industry foundation classes Web-based collaborative construction computer environment: WISPER", Automation in Construction, 10 (2000) 79-99
- [Chau, 20] K.W. Chau, Ying Cao, M. Anson, Jianping Zhang, "Application of data warehouse and Decision Support System in construction management", Automation in Construction, 12 (2002) 213 – 224

- [Songer, 21] A.D. Songer, B. Hays, C. North, "Multidimensional Visualization of Project Control Data", *Construction Innovation*, 4 (2004) 173 – 190
- [Microsoft, 22] Microsoft, "Microsoft .NET Frequently Asked Questions", [http://www.microsoft.com/net/basics\\_faq.mspx](http://www.microsoft.com/net/basics_faq.mspx) (Accessed November 2004)
- [Kirtland, 23] Mary Kirtland, "The Programmable Web: Web Services Provides Building Blocks for the Microsoft .NET Framework", *MSDN Magazine*, September 2000
- [Larsen, 24] Marcy Larsen, "Business Intelligence Within Reach", Microsoft Whitepaper, November 2002
- [Microsoft, 25] Microsoft, "Microsoft Business Intelligence Platform", <http://www.microsoft.com/business/productivity/decision/intelligence/default.mspx> (Accessed September 2004)
- [Larsen, 26] Diane Larsen, Euan Garden, "Data Transformation Services (DTS) in Microsoft SQL Server 2000", *MSDN Library*, September 2000
- [Microsoft, 27] Microsoft, "Reporting Services Product Overview", <http://www.microsoft.com/sql/reporting/productinfo/overview.asp> (Accessed, January 2005)
- [Microsoft, 28] Microsoft, "Office XP Tool: Web Components", <http://www.microsoft.com/sql/reporting/productinfo/overview.asp>
- [Ben-Moshe, 29] Tomer Ben-Moshe, "Microsoft Business Intelligence Portal—BIP", Applies to Microsoft Office XP, Microsoft Corporation, May 2003
- [Microsoft, 30] Microsoft, "Microsoft Office Solution Accelerator for Business Scorecards & Excel Reporting", <http://www.microsoft.com/korea/magazine/200407/newproduct/np.asp>