Lecture Notes in Computer Science, Vol. 2480, 2002, pp. 360-369

An Intelligent Decision Support System in Construction Management by Data Warehousing Technique

Ying Cao[1], K.W. Chau[2], M. Anson[2] and Jianping Zhang[1]

[1]Dept. of Civil Engineering, Tsinghua University, Beijing 100084, China
[2]Dept. of Civil & Structural Engineering, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk

Abstract. The integration of a Data Warehouse and a Decision Support System (DSS) can provide construction managers with sufficient information for decision making without interrupting daily work of an On-Line Transaction Processing system. In this paper, the concepts of the data warehouse, On-Line Analysis Processing and DSS are first reviewed. The method of creating a data warehouse is then shown, changing the data in the data warehouse into a multidimensional data cube and integrating the data warehouse with a DSS. An application example is given to illustrate the use of the Construction Management Decision Support System developed in this study. This prototype system can enable the right data to be tracked down and provides the required information in a direct, rapid and meaningful way. Construction managers can view data from various perspectives with significantly reduced query time, thus making decisions more efficiently. Moreover, the approach can be applied to other fields.

1 Introduction

Today, a construction company typically has some transaction processing systems, which run the day-to-day business of the company, and are updated continually throughout the day [1]. For instance, if some materials are delivered into the warehouse, the On-Line Transaction Processing (OLTP) will consistently make additions to the inventory. However, it is usually found in such systems that the construction process is a “temporary” and “specific” activity, which means the data of one project can seldom be used for another project. Is that the true situation? Probably not, because although construction products are ‘unique’, some similarities still exist between them and construction processes and management skills are typically common to all projects [2]. How to analyze the successes and failures of finished projects and how to use the existing data to analyze patterns and trends for new projects are the problems we have to face.

In addition, project managers, during the project control phase, need timely analysis reports to measure and monitor construction performance, attempting to take rectifying actions for any deviations in the performance. They also need timely analysis reports to assist in making long-term decisions [3]. It is found that most of the reporting and analysis time was spent on collecting data from the various systems before the analysis can be made. Managers want and need more information, but analysts can provide only minimal information at a high cost within the desired time frames [4]. In order to provide information for predicting patterns and trends more convincingly and for analyzing a problem or situation more efficiently, an integrated decision support system (DSS) designed for this particular purpose is needed.

A DSS provides information enabling users to analyze situations and make decisions [5]. This decision-making can be of a long-term strategic nature, such as analyzing event patterns over several years to prevent or reduce the rate of occurrence of a particular event. It can also be short-term and tactical in nature, such as reviewing and changing the time schedule for a particular part of a project. In this paper, the development of a prototype DSS employing the new ‘data warehouse’ technology incorporating large quantity of analysis information needed for both long-term and short-term construction management decision-making, is described.

The ‘Data warehouse’ is a new branch of the database ‘community’ developed in recent years [6]. It is a read-only analytical database that is used as the foundation of a DSS, with the purpose to ensure the appropriate data being available to the appropriate end user at the appropriate time. A data warehouse is a global repository that stores pre-processed queries on data, which reside in multiple, possibly heterogeneous, operational query base for making effective decisions. The contents of a data warehouse may be a replica of part of some source data or they may be the results of preprocessed queries or both. This method of data storage provides a powerful tool helping project organizations in making decisions. The data warehouse technique has been applied successfully to marketing and service environment [7].

The architecture of a data warehousing system allows a number of alternative ways to integrate and query (such as previous or projected) information stored in it. Thus, a data warehouse coupled with On-Line Analysis Processing (OLAP) enables project managers to creatively approach, analyze and understand project problems. The data warehouse system is used to provide solutions for construction problems since it transforms operational data into strategic decision-making information. The data warehouse stores summarized information instead of operational data. This summarized information is time-variant and provides effective answers to queries, such as “What are the supply patterns and trends of various construction materials?”, “How is the material consumption this year different from its counterpart last year?”, “How many accidents happened in the last ten years and how much did they cost?”, “What is the percentage increase in the cost of human resources during the last five years?”, “Did machine repair have any influence on construction progress? If so, what was the influence coefficient?” and so on. To extract this information from a distributed relational model, we would need to query multiple data sources and integrate the information at a particular point before presenting the answers to the user. It is time-consuming and is not online. In a data warehouse, such queries find their answers in a central place, thus reducing the processing and management costs.

Although DSS have been applied in construction management for several years, early systems such as management information systems, report-oriented systems and so on are often born with flaws. Firstly, they are not separated from transaction systems completely and the sharing of a database or data file slows down either transaction or analysis process. Secondly, because of the limitations of a relational database, users can only observe their data from flat views. Thirdly, these applications are all developed by computer specialists in information centers after lengthy data analysis, but sometimes not all the requirements of construction managers are embodied sufficiently. These problems could be solved in the Construction Management Decision Support System (CMDSS), developed in this study. The main characteristic of CMDSS is the separation of the analysis database (data warehouse) from the operational database, which renders the decision support process much faster. Another advance is the use of OLAP, which changes the data in a relational database into multidimensional cubes that could be observed from all perspectives. In addition, graphic visualization presents data from several kinds of views. Users could do a lot more on their own without computer experts preprogramming everything for them. As such, construction managers can make decisions more efficiently.

2 Architecture of CMDSS

The data models for designing traditional OLTP systems are not well suited for modeling complex queries in a data warehouse environment. The transactions in OLTP systems are made up of simple, pre-defined queries. In the data warehouse environments, queries tend to use connections between tables and have a longer computation time. This kind of processing environment warrants a multidimensional data model, a new perspective on data modeling. The conceptual multidimensional data model can be physically realized in two ways, (1) by using a trusted relational database approach (star schema/snowflake schema) or (2) by making use of a specialized multidimensional database. For reasons of its clarity, convenience and rapid indexing ability, the ‘star’ schema is adopted here [8].

A star schema is a specific type of database design used to support analytical processing, which includes a specific set of de-normalized tables. A star schema contains two types of tables: fact tables and dimension tables. Fact tables contain the quantitative or factual data about a construction management entity. Dimension tables are smaller and hold descriptive data that reflect the dimensions of an entity. SQL queries then use predefined and user defined links between the fact and dimension tables within the star schema, with constraints on the data to return required information.

In CMDSS, a typical material inventory model with sample material inventory fact table and five dimension tables is shown in Figure 1. These dimension tables are connected with the fact table by foreign keys (FK), which can keep all the views coherent. Besides the inventory star schema, several other star schemas are designed in our system, including material issuing, material balance, material use, machine cost, machine use, machine repair, human source use, salary, progress, disqualification, event, etc.

Apart from the data model design, several other necessary steps include: (1) data extraction from the source systems, databases and files; (2) data integration with the source systems; (3) data loading into the data warehouse; (4) data transformation into the format of the front-end tool. In CMDSS, the “Import and Export Data” tool is used to integrate data from distributed OLTP databases, files, etc.

The OLAP tool is employed to transform the fact table and dimension tables in the star schema designed above into a multidimensional cube that can be further explored by the front-end tools, such as Visual Basic, MS Access, MS Excel. Microsoft OLAP Services is based on and tightly linked to relational databases, yet being a real multidimensional information system. The OLAP structures are a valuable feature because many important analyses are difficult or impossible to phrase in SQL using tabular structures. Microsoft OLAP Services provides a relatively well-optimized solution [9]. For example, one characteristic of most OLAP applications is the need to provide fast access to aggregated source data. Pre-calculating all possible aggregations can lead to a tremendous increase in the storage requirements for the database, while calculating all aggregations on each occasion makes for a slow query response time. The approach taken by OLAP Services is to pre-calculate some of the possible aggregate data values, and to leave any remaining aggregation and all other calculations to be completed at query time.

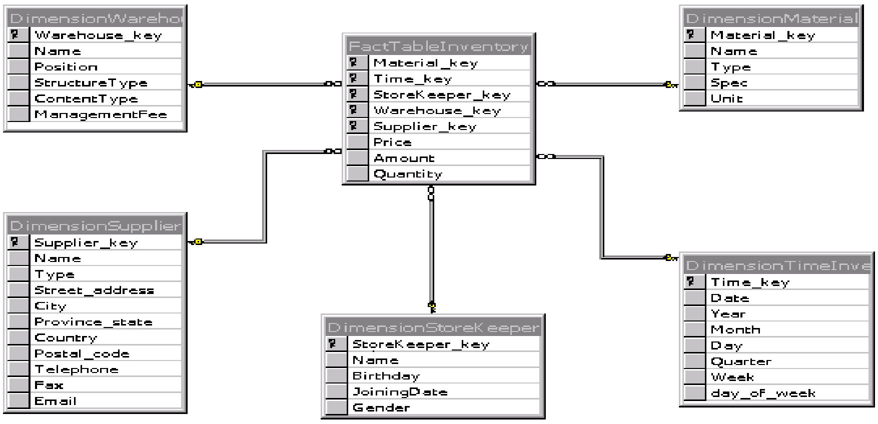
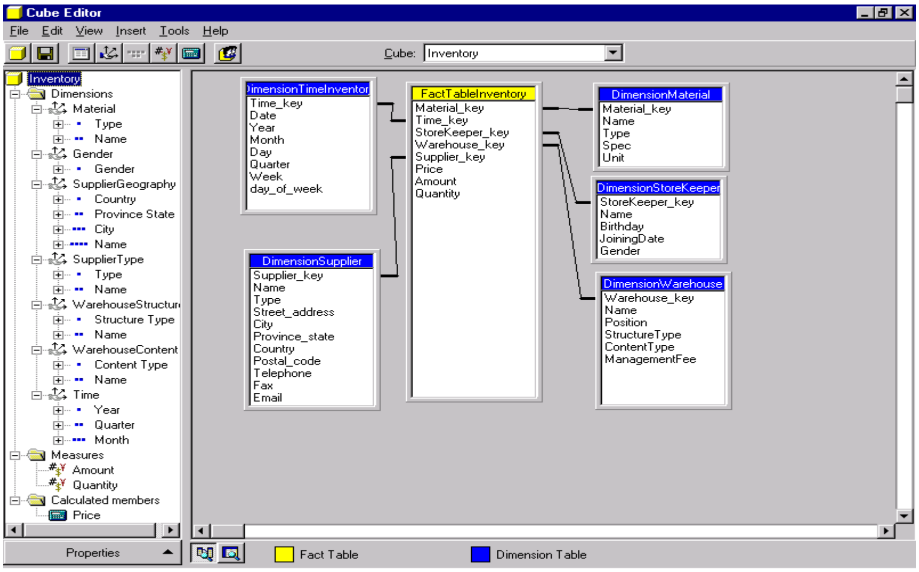


Fig. 1. Material Inventory Star Schema

The basic OLAP structures that are employed include cubes, dimensions, measures, hierarchies, levels, and cells. These, taken together, define the logical structure of an OLAP database. A data cube is a structure for housing multidimensional data. Measures are the data that we wish to analyze, while dimensions define the organization of these measures [10]. For instance, our data warehouse contains an inventory table that has fields for warehouse, time, material, supplier, storekeeper, price, quantity and total amount. We generally analyze price, quantity and total amount by warehouse, time, material, supplier and storekeeper. In this case, price, quantity and total amount will be our measures, and warehouse, time, material, supplier and storekeeper will each be a dimension. The elements of a dimension are called members. The path to organize members in a cube is called hierarchy. For instance, the time dimension may be organized in two hierarchies: natural time hierarchy and fiscal time hierarchy. In the former one, time may be organized in year, quarter, month and date. While in the latter one, time may be organized in fiscal month, fiscal week. A level refers to a group of related members which share a common meaning. For example, a level construct named ‘month’ may contain all of the month-level members in a time dimension. Each unique intersection composed of one member from every dimension in the cube is called a cell. For instance, the intersection of July (member of Time) and Hong Kong (member of Geography) constructs one cell, and the value of the cell could be either measure, such as price, quantity or total amount.

All dimension levels and cube measures need to correspond to columns of tables, views, or queries. They can be in many different tables or all in one table, so long as dimension tables and fact tables can be joined in a single query. OLAP Services uses a highly declarative linkage among dimension, cube structures and Relational Database Management System tables. Once the links are created, OLAP Services will form all queries on the linked tables and manipulate all query results. Many cubes about material, machine, human source, progress, quality and event are created in our system on the basis of the star schemas designed before. New multidimensional cubes can be added at any moment by the users as the need arises. An ‘Inventory’ cube created in our system is shown in Figure 2. The major operations that could be done on OLAP cubes are Selection, Roll-up, Drill-Down and Slice, through which we can view data from all perspectives and all levels [11].



Cube: Inventory

Inventory
- Dimensions
  - Material
    - Type
    - Name
  - Gender
    - Gender
  - SupplierGeography
    - Country
    - Province State
    - City
    - Name
  - SupplierType
    - Type
    - Name
  - WarehouseStructur
    - Structure Type
    - Name
  - WarehouseContent
    - Content Type
    - Name
  - Time
    - Year
    - Quarter
    - Month
- Measures
  - Amount
  - Quantity
- Calculated members
  - Price

)imensionTimeInventor: Time_key, Date, Year, Month, Day, Quarter, Week, day_of_week

FactTableInventory: Material_key, Time_key, StoreKeeper_key, Warehouse_key, Supplier_key, Price, Amount, Quantity

DimensionMaterial: Material_key, Name, Type, Spec, Unit

DimensionStoreKeeper: StoreKeeper_key, Name, Birthday, JoiningDate, Gender

DimensionSupplier: Supplier_key, Name, Type, Street_address, City, Province_state, Country, Postal_code, Telephone, Fax, Email

DimensionWarehouse: Warehouse_key, Name, Position, StructureType, ContentType, ManagementFee

Fig. 2. Inventory Cube in Microsoft OLAP

Before the design of a DSS, some fundamental questions should be first understood. How are reports and analyses shared between users? How are structured navigation paths or command buttons created? Is there a query controller to limit the allowable elapsed time run for a query or to limit the total number of rows that can be returned? Is there an ability to run a query during off-peak hours to save costs? … These questions are quite important for deciding the aim and the direction of a DSS, which also means the success of it.

Since the DSS is designed to allow analysts to extract information and historical data quickly and easily, the system development life cycle for DSS is quite different from an OLTP system. The focus of a DSS is data, not construction processing and their associated functionality. This lack of domain functionality implies a much faster development life cycle.

The DSS application provides predefined reports created by a front-end tool to accommodate the need for different levels of users to have prebuilt reports to begin their analysis. So, the general data access processes are visualization of the data warehouse, formulation of the request, processing the request, and presentation of the results. After the data warehouse design and OLAP transformation, two steps are left to create a DSS application. The first is to design the front–end interface. The second is to generate codes to access and navigate metadata to obtain information on the data in the warehouse, and link it together with the front-end interface.

The level of users should always be a major consideration in designing the DSS interface. Facing different users, different type of DSS interface should be designed. Generally, we may face two distinctly different levels of users: (1) the experienced users who develop ad hoc queries using parameters. These users were also trained on the data model; (2) the novice or casual users, who are most comfortable in a point and click environment, where icons are employed instead of queries. For the above reason, two different interfaces were designed for our CMDSS. The first is designed for the experienced users. They can select parameters in the parameter dialog box and view the data more flexibly. The second interface is designed for the novice who can just click and view the predefined information directly. The selection interface is shown in Figure 3.

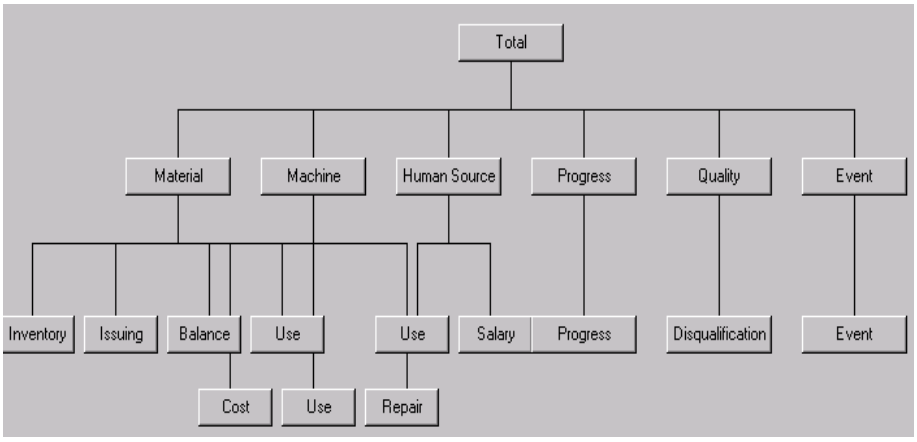


Fig. 3. Predefined Information Selection Box

Another characteristic of a DSS is that data returned from a request can be displayed in a wide variety of ways. Here data reports and some graphical views are explored to view the data and assist users in gathering insights from the data. Once a data report is presented, a wide range of capabilities may allow modification of the display, including:

changing the axis of the report (Rows and columns), changing the sort order of the results, adding subtotals and grand totals at appropriate breaks in the reports, formatting of fonts, styles, sizes, and colors. Graphical display of information allows easy detection of trends and anomalies. A wide variety of graph types are available here, including line, pie, bar, area and so on. With a graphical display, users may be able to change graph type, axis labels, colors and titles.

After the design of the interface, the next step of our work was to generate codes linking the data warehouse and the interface together. In the novice interface, once a user presses a button, a Precoded SQL statement will be passed directly to the Multidimensional Database Management System (MDBMS) and the appropriate answer set will be displayed on the interface by the prediction function. But in the interface for experienced users, once a user has formulated a request, our system can translate the end user’s request to generate the appropriate SQL statements, get the result set from the MDBMS and display it in a variety of views. Because of its good connections with databases, Visual Basic 6.0 was adopted here to develop our CMDSS, using also ActiveX Data Objectives Multidimensional as a vehicle for communication. The following code is a sample used to display the value of a cell in a grid.

```
For intColum = 0 To Cellset.Axes(0).Positions.Count –1
For intRow = 0 To Cellset.Axes(1).Positions.Count –1
 Grid.Column = intColumn + intFixedColumns
 Grid.Row = intRow + intFixedRows
 Grid.Value = Cellset(intColumn, intRow).FormattedValue
Next intRow
Next intColumn
```

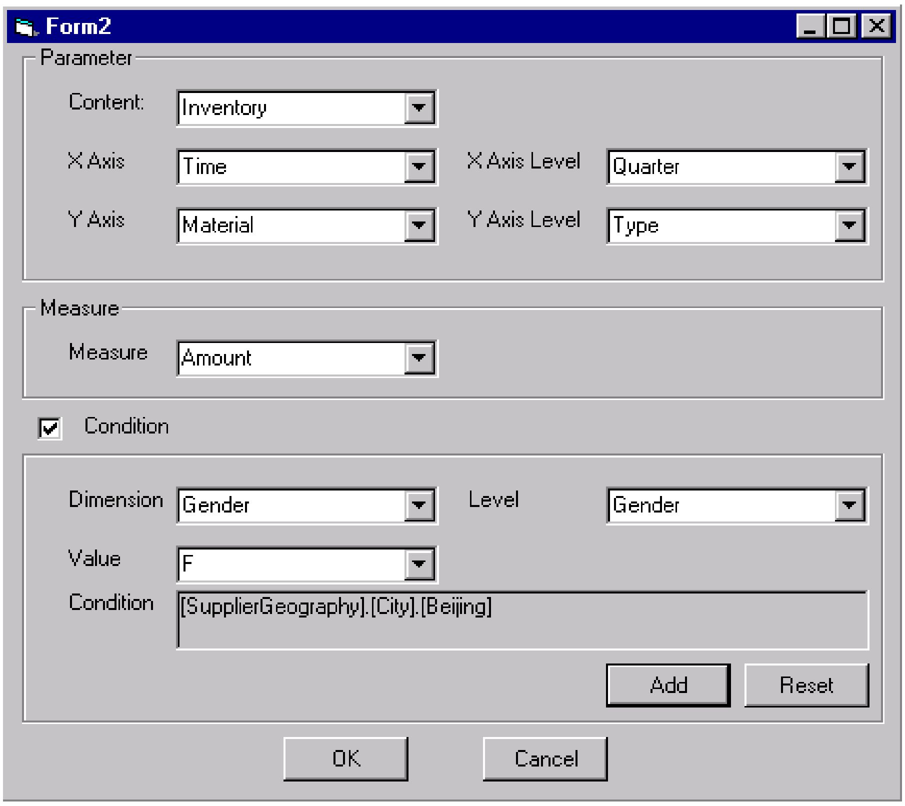
Beyond these fundamental steps, there are other advanced capabilities that many construction organizations require to support the full decision-making process. These may include: exception reporting (alerts are messages that appear when user defined conditions are not met or when there is a problem), drill-down (users have the freedom to “drive off” the existing report and retrieve information that may lie along, above, or below the current level of detail. Drill-down can be done from a report or a graph), data surfacing (we can keep the report layout constant but change the constraints. For example, a material-consumption trend report for concrete in Hong Kong, changed to a material-consumption trend for steel in Beijing), ranking (review information that is ranked on one or more columns), and automation (mechanisms are in place to schedule recurring analysis at a specified time).

3 Application Example

The Hong Kong Polytechnic University Student Dormitory construction project was chosen as an application example. There are several steps in using CMDSS:

- ➢ Choosing the interface: There are two kinds of interface in our system. As explained above, one is for novices and the other for experienced users. Here we choose a self-defined interface, which is designed for experienced users.

- Choosing display views: There are several views available to display the query result, including line chart, pie chart, bar chart, area chart and report file. Users can choose any of them.



Form2

Parameter

Content: Inventory

X Axis: Time — X Axis Level: Quarter

Y Axis: Material — Y Axis Level: Type

Measure

Measure: Amount

☑ Condition

Dimension: Gender — Level: Gender

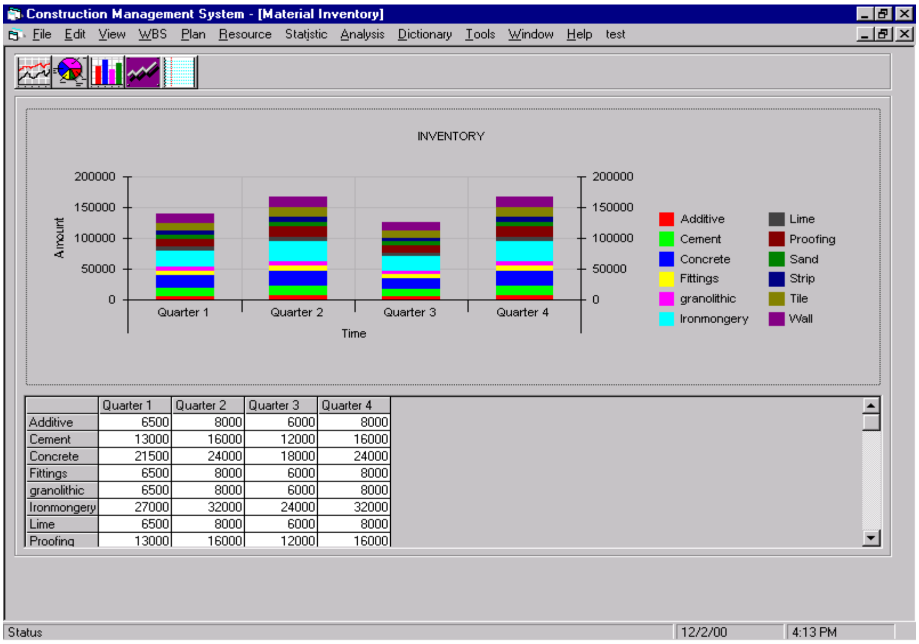
Value: F

Condition: [SupplierGeography].[City].[Beijing]

Fig. 4. Parameter Selection

- Input parameters: Some parameters are entered before the query result is viewed. Here we choose inventory as an observing cube, and choose the Time dimension (Quarter level) as the X Axis, the Material dimension (Material Kind level) as the Y Axis and the Amount as measure. Users can also attach some conditions to the query. Conditions can be selected in the combo-box and added by ‘Add’ button or cleared by ‘Reset’ button. In this example, we want to determine the total amount of the materials stored in the warehouse whose storekeepers are female with materials from suppliers in Beijing. This condition is displayed in the condition box as “[SupplierGeography].[City].[Beijing], [Gender].[Gender].[F]”. The parameter selection dialog box is shown in Figure 4.
- Result display: After entering the parameters, results will be displayed on the interface. An Inventory cube, which can be observed in several views in the DSS, is shown in Figure 5. In the chart, X-axis is Time Dimension and Y-axis is

Total Cumulative Amount together with contributions from different material types in four quarters of a year. The legend on the right of the chart displays the colors representing different material types. In the grid shown below the chart, the values of the amount of each material type at each quarter of the year are given. The same information can be observed in four different types of charts: Line chart, Pie chart, Bar chart and Area chart. Using this information, the managers can then formulate an appropriate inventory decision or warehouse storage strategy.



	Quarter 1	Quarter 2	Quarter 3	Quarter 4
Additive	6500	8000	6000	8000
Cement	13000	16000	12000	16000
Concrete	21500	24000	18000	24000
Fittings	6500	8000	6000	8000
granolithic	6500	8000	6000	8000
Ironmongery	27000	32000	24000	32000
Lime	6500	8000	6000	8000
Proofing	13000	16000	12000	16000

Fig. 5. Material Inventory observed in Form of Bar Chart

4 Conclusions

This paper describes the development of a prototype decision support system, CMDSS, employing the new ‘data warehouse’ technology incorporating a large quantity of analysis information needed for both long-term and short-term decision-making in the construction management process. Due to the integration of the data warehouse with an OLAP, CMDSS is advanced in the following aspects: CMDSS enables insight to be gained into the factors having impacts on construction management activities that will help managers in making decisions to improve management performance. CMDSS is interactive. Users can interact with the computer so that the users can constantly refine the view of data to pursue various ways of thought. CMDSS provides extremely fast response to queries. CMDSS is multidimensional. Users can view figures from multiple perspectives and can also choose different view angles. In Short, CMDSS is able to assist project managers by providing accurate and timely information for construction decision-making. The integration of a data warehouse and a DSS is a promising way to solve decision-making problems during construction management process. Moreover, the approach can be applied to other fields.

5 Acknowledgement

This research is supported by the Research Grants Council of Hong Kong (PolyU5060/99E) and the Natural Science Foundation of China (No. 59778055).

References

1. Sarda, N.L.: Temporal Issues in Data Warehouse Systems. Proceedings of the International Symposium on Database Application in Non-traditional Environments. (1999) 27-34
2. Yang, Jyh-Bin, Yau, Nie-Jia: Application of Case-Based Reasoning in Construction Engineering and Management. Proceedings of the Third Congress of Computing in Civil Engineering held in conjunction with A/E/C Systems. (1996) 663-669
3. Vanegas, J., Chinowsky, P.: Computing in civil Engineering. American Society of Civil Engineers (1996)
4. Dyche, J.: e-Data Turning Data into Information with Data Warehousing. Addison-Wesley (2000)
5. Poe, V., Klauer, P., Brobst S.: Building A Data warehouse for Decision Support. Prentice Hall (1998)
6. Samtani, S., Mohania, M., Kumar, V., Kambayashi, Y.: Recent Advances and Research Problems in Data Warehousing. Advances in Database Technologies. Proceedings of ER'98 Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spation-temporal Data Management. Singapore (1998) 81-92
7. Marks, W.T., Frolick, M.N.: Building Customer Data Warehouses for a Marketing and Service Environment: A Case Study. Information Systems Management **18(3)** (2000) 51-56
8. Corey, M., Abbey M.: SQL Server 7 Data Warehousing. McGraw-Hill (1999)
9. Peterson, T., Pinkelman, J.: Microsoft OLAP Unleashed. SAMS (1999)
10. Thomsen, E., Spofford, G., Chase, D.: Microsoft OLAP Solutions. Wiley Computer Publishing (1999)
11. Vassiliadis, P., Skiadopoulos, S.: Modelling and Optimisation Issues for multidimensional Databases. Proceedings of the 12th International Conference on Advanced Information System Engineering. Stockholm (2000) 482-497