Simulation of an Assistant Mode in the Data Warehouse System

Yung-Hsin Wang
Dept.of Information Management
Tatung University
40 Chungshan North Road, 3rd Sec.
Taipei 104, Taiwan, R.O.C.
E-mail: ywang@mis.ttu.edu.tw

Mo-Mau Su
Tainet Communication System Corp.
No. 25, Alley 15, Lane 120,
Sec. 1 Nei-Hu Rd., Taipei,
Taiwan, R.O.C.
Email: samuel@tainet.net

Abstract

The data in a data warehouse can be obtained from a variety of sources, which have many formats and need to b integrated. The data warehouse thus serves as a repository for vast amount of various data that may be updated in a cycle time. The changes of source databases can be critical to the correctness of the query responses provided by th data warehouse. This paper presents an assistant model for the data warehouse, which accepts and stores the source updates in temporary data storage until the data update cycle arrives. At that time, data are then integrated and updated in the data warehouse. The assistant is modeled to avoid the data inconsistency and to release the burden of the data warehouse. The SES/workbench, a graphical modeling and simulation tool, is used to construct models necessary for the proposed data warehouse system. Performance comparisons are made between the data warehouse with and without the assistant. Simulation results show that the data warehouse with the assistant model outperforms the one without.

Keyword: data warehouse, database, modeling, simulation.

1. Introduction

The data warehouse is a blend of technologies and components aimed at effective integration of operational database into an environment that enables strategic use of data. It is an architectural construct of an information system providing user with current and historical decision supports information th at is hard to access or present in traditional operational data stores. Because of the vast amount of data in the data warehouse, we must spend a lot of time to extract suitable data by querying the data warehouse. The workloads are query intensive with mo stl ad hoc, complex queries that can access millions of records and perform a lot of scans, joins, and aggregates. Thus, query throughput and response times are more important than transaction throughput.

The original approach to updatin a data warehouse is to gather all of the updated information from the source databases at some predetermined times, preferably durin the last active period of the day and transfer the information to the data warehouse, and then perform data incorporation and view updates immediately. But the difficulty is that the source databases are changing rapidly. The changes made to the source databases can be critical t the correctness of the query responses provided by the warehouse.

The following are the four methods follow the techniques described in Inmon [6] to obtain the data from the data warehouse. They are:

(a) Modifying the application of source databases to provide the update data queried for the data warehouse.
(b) Creating a log file or a special file to record the required information of previous access and extraction.
(c) Maintaining a snapshot of the source databases and comparing it with the current contents of the source databases. If there are any changes in the source databases, the changed data are extracted and forwarded to the data warehouse.
(d) Scanning the data that has been timestamp ed. After data scanning, integration is the most important aspect of a data warehouse.

When data passed from the application-oriented operational environment to the data warehouse, incons istencies and redundancies may occur and should be resolved to provide an integrated and reconciled view of data.

The data warehouse is one way to store the data of the enterprise. Since there are many departments of the enterprise dispersing in different country and area the source files can be in various formats and we should integrate these data and analyze the information according to the data sources or the matedata [4]. After change of new data, we have to spend a lot of time to assimilate and update these new data into the data warehouse, and after the data transmission, we have to concern with data consistency between the data warehouse and source databases. The data warehouse should also integrate various forms of data and store them. We must defin e a unique form that will make it available to the user communit [9]. Although the third method described above has a serious problem of being too complex and requires an excess amount of resources, we can get the information without destroying the source databases. Thus, we attempt to use the similar concept of the approach to implement our system model to improve the performance.

As already known in the computer system, a cache is used to improve the performance of the system. Likewise, we propose an assistant model to prestore the data from the data sources and provide the user to query the data warehouse. The prestored data is also like a snapshot mentioned above. The two snapshots—before and after—are serially compared to each other to determine the updated data (metadata). The existence of the assistant model will reduce the distance between the data sources and data warehouse. Furthermore, the assistant can share responsibility of the data warehouse to release the workload of the data warehouse. Thus, our goal of this paper is to add an assistant to the data warehouse to provide users a faster response to any query submitted to the data warehouse, thereby the faster updated and consistent data can be obtained.

2. The Data Warehouse Architecture Overview

A data warehousing architecture is a way of representin the overall structure of data, communication, process ing and presentation that exist for end user computing within the enterprise. The architecture is made of a number of interconnected parts

- Operational database and external database: The purpose of the operational database is to address the need of users, particularly clerical and operational managers, for an integrated view of current data [5]. We refer to information that comes from outside an organization as external data [1].
- Data Warehouse: A data warehouse is just another database. What sets it apart is that the information it contains is not used for operational purpose, but rather for analytical task [12].
- Data Marts: A data mart might, in fact, be a set of denormalized, summarized, or aggregated data [2].
- OLAP Servers: On-line analytical processin (OLAP) is an application, not intrinsically a data warehouse or a database management system. OLAP is becoming an architecture that an increasing number of enterprises are implementin to support analytical application and provides the managers with the t ools they need t analyze large amount of data effectivel [13].
- Metadata Repository: Metadata is data about data [3]. It contains the location and description of warehouse system components.
- Tools: These tools are the application that provides the human readable form and can help the user know the data's meaning in the data warehouse.

A typical data warehousing architecture is shown in Figure 1 [11]. We begin extracting data from multiple operational databases and external sources, then transforming and integrating the data to the data warehouse. After integration data will be loaded into the data warehouse.

3. The Proposed System Model Overview

As has been described in the previous section, the data warehouse can accept the data of interest from multipl e source databases directly into its table. In order to improve the performance of the data warehouse, we will add a n assistant model into the overall data warehouse system. The function of the assistant model is like a cache in the syste and it will be added between the actual data warehouse and the source databases. If there is any update of the data, the updated information will be transported into the assistant. Then the assistant will send the update data to the data warehouse when an update cycle time arrives. Figure 2 shows the data flow of the system. The component of the system will be described subsequently.

3.1 The Overall System Model

We shall now describe in this section how the system will be formed. There are three models in the overall system:

- Assistant model: the assistant model is to act as a cache that holds area between the data warehouse and the data sources to assimilate the data from the source databases and integrate them.
- Data warehouse model: it stores all data and provides data information in the whole system. It also provides data analysis to the manager for decision making [8].
- Report generation model: it will provide the outcome after data analysis.

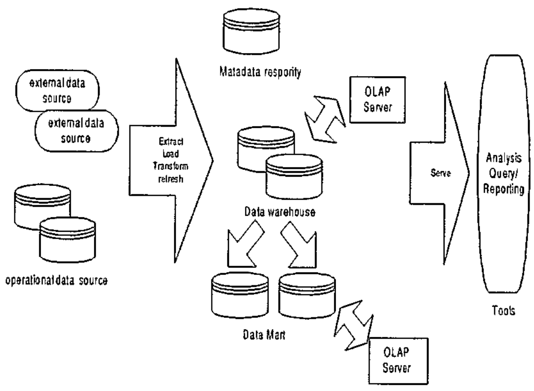

Figure 1. Data warehouse architecture

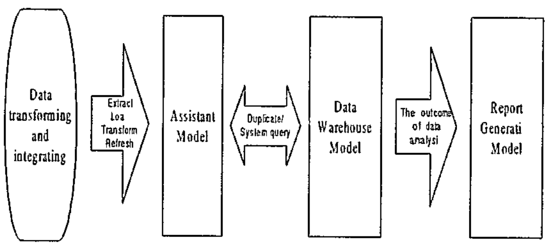
Data transforming and integrating
Extract Loa Transform Refresh
Assistant Model
Duplicate/ System query
Data Warehouse Model
The outcome of data analysi
Report Generati Model

Figure 2. Data flow in the purposed system

3.2 The Assistant Model

The assistant model is a coordinator between the data warehouse and the source databases. Data from the source databases are transferred and integrated to produce the data information (i.e., the update data). This update data are then sent to and placed into the assistant model. The assistant model just stores the delta information for later data warehouse update and should not change the structure of the data warehouse. There are five components in the assistant model as shown in Figure 3. They are:

- The Update Data Queue: the queue is to store the new update data from the data transport.
- The Storage Model: the model is to store the new update data that came from the Update Data Queue. It will then send the new update data to the data ware house. The following are the three components in the storage model.
  - Information Manager: it is to manage and store th new update data in the Data Information Storage.
  - Data Information Storage: it is to store the ne update data information from the source databases.
  - Space Manager: it is to manage the storage space of the assistant model.
- Transaction Manager: the manger is to manage that when the new update data should be transferred to the Assistant and then to the DataWarehouse.

We shall now introduce these components individually in what follows.

The Update Data Queue

The Update Data Queue is a model that waits for the new update from the source databases. When the new update data arrives, it will be put into the queue. Then, we use the FIFO (first in first out) strategy to handle the update data from the source databases. The Transaction Manager will control the data transfer from the queue to the Information Manager.

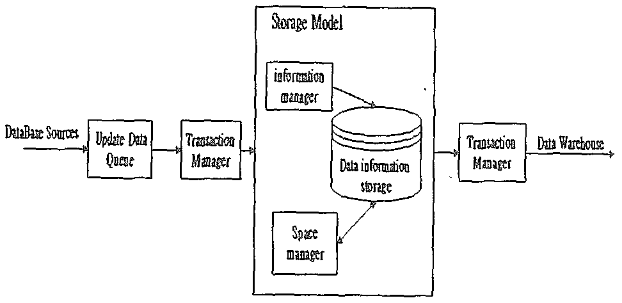

Figure 3. The assistant model

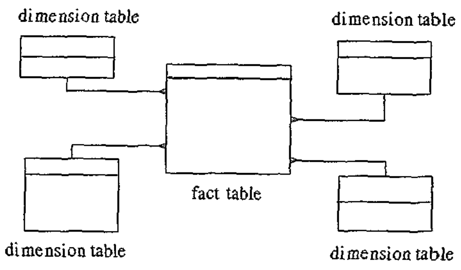

Figure 4. The star schema

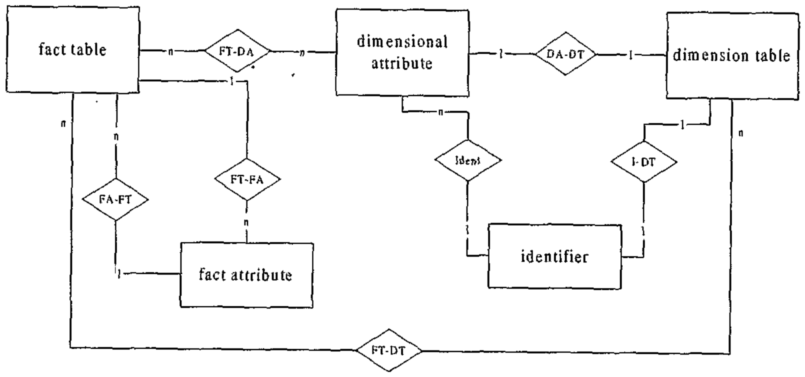

Figure 5. The E-R model of the metadata in the Information Manager

The Information Manager

We store our data in the data warehouse with the star schema. The star schema as shown in Figure 4 takes its name from the star configuration that is represented by this approach [7].

The Information Manager uses the same schema as the data warehouse, i.e. the star schema, to manage the update data from the source databases. When receives the update data, the manager will assimilate the data's information from the source database and map them to the data warehouse. The detailed information of the schema in the Information Manager is described in Figure 5.

Three basic classes of the star schema exist in the original model of the Information Manager: fact table, dimensional attribute, and dimension table. Value sets are associated with fact attribute and dimensional attribute, with the usual meaning.

As can be seen in Figure 5, an identifier may be attached to several possible dimensional attributes, and an identifier may be attached to a dimensional table. One fact table ma have several fact attributes, and a fact attribute may be attached to several possible fact tables.

A metadata contains all other databases information in our system. We use the star schema described above to represent the metadata in our system. If there is any change in one of the databases, the metadata records the change. The information of mapping between the source databases and assistant is processed by the Information Manager and stored in the Data Information Storage.

The Data Information Storage

The Data Information Storage is the model to store the update data information. It provides the space to store the delta information of update data. If the cycle time of the data warehouse's update is up, the Data In formation Storage sends the data to the data warehouse.

The Space Manager

This model works to maintain the capacity of the assistant model. As already indicated, the purpose of the assistant model is to act as a cache and hold an area between the actual data warehouse and source databases. The data in the assistant model is the information of some new update data. There is limit space for the above purpo se. So, the Space Manager has to control the space of the Data Information Storage.

The Transaction Manager

The system efficiency to provide data complete and recent is very important to users. So, the Transaction Manager is to manage the process flow of our assistant model. We can control data refreshment by the Transaction Manager. We define a *data update time*, δ, that controls the update time of the data to be stored in the assistant model. If the period is equal to δ, the Transaction Manager will order the source databases to send the data into the assistant model. We also define second data update time, δ_2, that controls the data from the assistant to the data warehouse.

3.3 The Data Warehouse Model

The purpose of the data warehouse model is to analyze the data from the user query. The data warehouse also stores the data and metadata. So, it can provide the user data querying and data searching. The model has three submodels: the Query Model, the Metadata Manager, and the Storage Model. We shall now describe the three models in what follows.

The Metadata Manager

The Metadata Manager is to store the metadata of the data warehouse. The data warehouse solves the user query according to the metadata. Any change of the source database will be recorded in the assistant's Data Information Storage. When the cycle time of the update data is up, the Transaction Manager orders the Data Information Storage to send the data to the Metadata Manager. Thus, the Metadata Manager always receives the newest data.

The Storage Model

The model is to store the data in the data warehouse. It can store the operational data and information of the data (metadata). When the data is stored in the Storage Model, its information will be recorded in the metadata, a entity in the metadata has the data information of the data in the data warehouse. If the data is queried next time, the query processor in the Query Model can find the data in the Storage Model via the metadata. In this way, the response time of the user query will decrease. If the data is deleted from the Storage Model, the space that deleted data occupied must return to the Storage Mode l. Also, the data information in the metadata has to be deleted.

The Query Model

The purpose of the Query Model is to solve the query addressed by the user. There is a querying processor in the Query Model, which can process the query received from the c lient application. When the user query arrives, the processor will base on the condition of the user query to search or compute the data from the metadata. There is also a metadata in the data warehouse to record the information of where the data is. The m odel can then find the suitable data according to the metadata to resolve the user query.

3.4 The Report Generation Model

The data warehouse should provide the result of the user query. It shows the analysis of the queried data for users. I should also produce the form that users can understand and utilize to make a good decision.

4. Model Design and Simulation Results

4.1 The Simulation Model

This section presents the modeling and simulation studies designed to evaluate our data warehouse system with the assistant model. The purpose of this simulation study is to compare the response time of data updating in a data warehouse with the proposed assistant model and that in the one without an assistant model.

We shall construct the models according to the data warehouse architecture. A query processor model will be created to manage the query. The simulation will start from the query In the data warehouse system with an assistant model, we also construct the models according to the data warehouse system described in Section 3.

In our simulation design, there are three main models for the whole data warehouse system: the Entire System model, the Data Warehouse model, and the Assistant model. We use SES/*workbench* [10], a graphical modeling and simulation tool, for all the model construction and simulation process

Our simulation model has one data source (to simulate multiple source databases) and one data warehouse model. Figure 6 is the data warehouse system with the Assistant model. All node functions in Figure 6 are explained in Table 1. In Figure 6, we can see that the data trigger inference engine is built in the Entire_system model. Before the data are sent into the assistant model, the transaction manager will control when the data is to go to the assistant model. If the update cycle time is up, the data will be sent into assistant model. Otherwise, the data will be held in the transaction manager until the update cycle time is up. Once the data is in the assistant model, the data information manager will use the new update data to modify the metadata. And, the space manager will control the memory allocation. When the update time for the data warehouse is up, the update data will be sent into the data warehouse (Figure 7) and will be processed by the metadata manager which maintains and stores the metadata in the storage model of the data warehouse. All node functions in Figure 7 are explained in Table 2.

In the data warehouse submodel, there is a user query source node to issue the user query. The user query will be sent into the query model and the query processor for processing. Figure 8 is the assistant model whose node functions are explained in Table 3. In our proposed assistant model, two tasks must be handled. One is user query and the other is data update. There is a flow control, which controls the transaction flow of the user query and update data from the data-source. For update data, the data information manager from the data information storage, which stores the metadata, must search it. If the update data is found in the data information storage, the information manager will modify the data information storage accordingly. If it is not found, a space must be allocated for its storage. The response time of such data update is determined by whether the data is already there. On the other hand, the assistant must handle the transaction flow of the user query.

4.2 Simulation Results and Analysis

The interesting performance indices of our simulation are: (1) The management performance of the update data and user query in the data warehouse . (2) The performance effect on the data warehouse of adding the assistant model. Parameters in Table 4 are used to calculate the response time in the flow path of the data handling. When a data transaction from the data sources arrives at the data warehouse, the job[jobno].ariv_time can record the data transaction's arrival time. Then, the data will be handled b the assistant model and be recorded into the metadata. After the data transaction is finished, its departure time w ill be recorded. We can use the following equation to determine the response time of the update data

Data response time = transaction departure time – transaction arrival time

When a query transaction flows from the query source to the data warehouse, we can record the time. Then, the query processor will handle the quer . The query processor will check the data metadata and find out the data. The following equation is used to show how much time the query spends:

Query response time = query completion time – query arrival time

We change the source external data and user query arrival rate to simulate the system model. We should notice that the user query's arrival rate has to be smaller (i.e., the mean interarrival time is larger) than that of the update data. If the user query rate is larger than the other, the data in the data warehouse may be empty and the query processor will not find any data for the user query. The parameters in Table 5 are used to simulate the data warehouse model. I our parameters, there is a function of expo(mean). The expo(mean) stands for an exponential distribution with the given mean. The stream parameter is optional and allows user-defined random number generator streams to be used. The time unit of the user query rate is second. Table 6 and Table 7 show the simulation results of that without the assistant.

Now, we will simulate the data warehouse model with the assistant model to find out its update performance on and user query response time. Table 8 and Table 9 show the simulation results of that with the assistant.
From the results, we can see that the assistant model helps the data warehouse get higher performance on the update data process and user query process. Although the four cases of the sources result differently, all experiments show that the assistant model makes the data warehouse perform better. Figure 9 to Figure 10 further illustrate this point.

In sum, our simulation results show that the system with the assistant model can decrease the response time. Users can get the query result faster wit the assistant model added. Furthermore, users should be able to get the more update data in the new data warehouse system.

5. Conclusions

This paper has described a way for improving the data warehouse's performance. The major difference between our approach and others' is that we append an assistant model into the data warehouse system. The addition of this model provides the data warehouse architecture acceptin data from the source databases as they are being updated without the system cost. The assistant contains the metadata of source databases that removes the necessity for any active user queries from the data warehouse back to the source databases, thereby ensuring that the source databases are available to the u ser community for the highest possible percentage of time. It also allows the most current data contained in the source databases to be forwarded to the data warehouse with the new metadata caused by user queries. In order to provide a rapid response to user's query, the assistant model also has good handlin with the user query. When there is a user query, its transaction must wait behind the currently executing data update in the data warehouse. This way will ensure that the data are queried correctly.

This study also contributes to the field of modeling and simulation by demonstrating how the data warehouse system is modeled and simulated in the SES/*workbench*, a graphical object-oriented simulation software. Simulation models of the data wareh ouse with and without the proposed assistant model are constructed separately and performance comparisons are made between both systems. Simulation results show that our proposed model has better performance than the one without the assistant.

References

[1] C. Adamson and M. Venerable, *Data Warehousing Design Solutions*, New York: John Wiley & Sons, 1998.

[2] A. Berson and S.J. Smith, *Data warehousing, Data Mining, and OLAP*, New York: McGraw-Hill, 1997.

[3] M.S. Chen and J.H. Yu, "Data mining: an overvie from a datab ase perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, pp 866–883, 1997.

[4] R. Conine, "The data warehouse in the telecommunications industry," *IEE Network Operations and Management Symposium*, Vol. 1, pp. 205-209, 1998.

[5] B. Devline, *Data Warehouse: from Architecture to Implementation*, Addison Wesley Longman. Reading, Massachusetts, 1997.

[6] W.H. Inmon, *Building the Data Warehouse*, Second Edition, Vol. 1, New York: John Wile & Sons, 1996.

[7] S. Kelly, *Data Warehousing in Action*, New York: John Wiley & Sons, 1997.

[8] W.J. Labio and H. Garcia-Molina, "Efficient Snapshot Differential Algorithms for Data Warehousing," Technical Report, Stanford University, Palo Alto, CA, 1996.

[9] B. Panda and W. Perrizo, "Reducing response time in a distributed data warehouse system," In *Proceedings of IEEE Southeastern'97 Engineering New Century*, pp. 48–51, 1997.

[10] SES, Inc., *SES/workbench Technical Reference, Release 3.1*, Scientific and Engineering Software, Inc., Austin, TX, February 1994.

[11] C. Surajit and D. Umes h, "An Overview of Data Warehousing and OLAP Technology," *ACM SIGMOD* record, Vol. 26, No. 1, March 1997.

[12] X. Zhang, "Data Warehouse Maintenance under Concurrent Schema and Data Updates," In *Proceedings of the 15th International Conference on Data Engineering*, pp. 490-499, 1999.

[13] Y. Zhao, K. Tufte and J.F. Naughton, "On the Performance of an Array -Based ADT for OLAP Workloads," Technical Report, Department of Computer Science at University of Wisconsin, Madison, May 6, 1996.

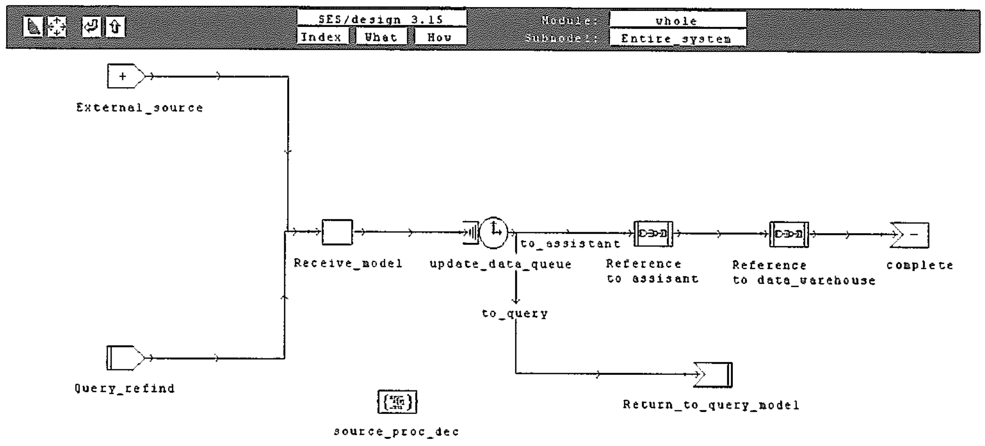

Figure 6. Simulation model of the Entire System with the Assistant

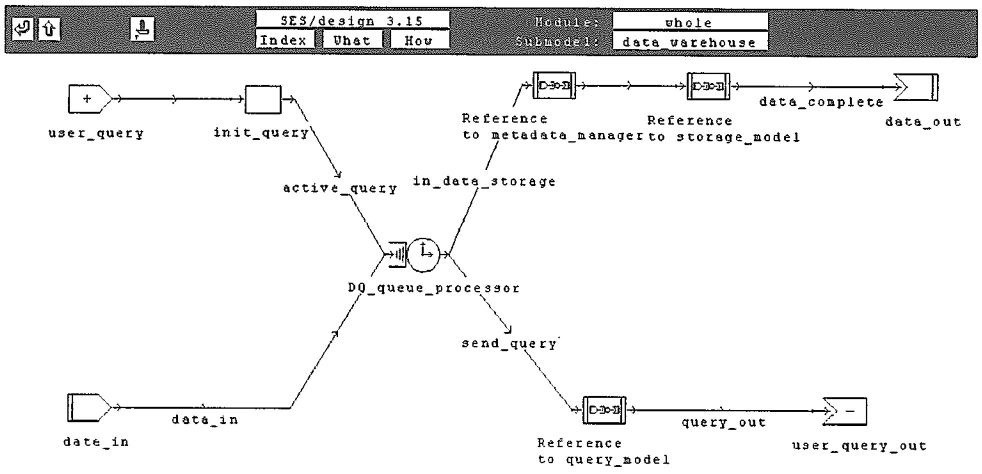

Figure 7. Simulation model of the DataWarehouse

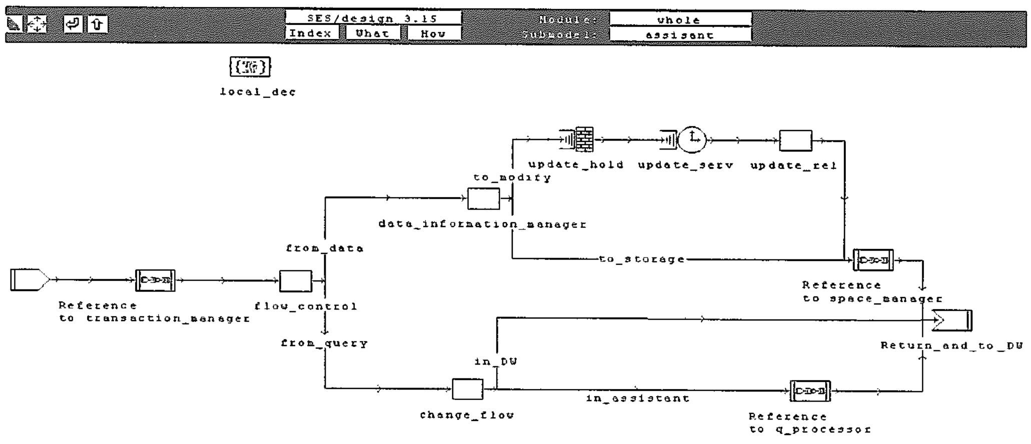

Figure 8. Simulation model of the Assistant

Table 1. The functionality of nodes in the Entire_system model with the assistant

Name	Function description
source_proc_dec	Declare variables used in this system model
External_source	To randomly issue the external data
Query_refind	To re-find the queried data of the source database (actually to insert the query data)
Receive_model	To receive the data from the data source or the query not found, and initialize the data table
update_data_queue	To decide to where the coming data will go, i.e., to the assistant model or to the quer processor
Reference to data_warehouse	Enter the data warehouse submodel
Return_to_query _model	Go to the query_model submodel
Reference to assistant	Enter the assistant submodel

Table 2. The functionality of nodes in the data warehouse model

Name	Function description
user_query	The source of the user query which is randomly issued
init_quer	The node is to initialize the user quer
data_in	The update data from the queue or the assistant
DQ_queue_processor	To queue the user query and data from the data source, and then dispatch them appropriatel
Reference to storage_model	Send the data to the storage_model for data storing
Reference to query_model	Send the query to the query_model for processing
Reference to metadata_manager	Send the update data to the metadata_manager for processing
Reference to transaction_manager	The transaction_manager submodel here controls the cycle time that data are sent to the data warehouse
user_query_out	The user query is completely processed

Table 3. The functionality of nodes in the assistant model

Name	Function description
local_dec	Declare variables used in this simulation model
data_information_storage	The space to store the new update data information
Reference to transaction_manager	The transaction_manager submodel have controls the cycle time that the data are sent t the assistant model
flow_control	Control the data fling into the assistant model. If the data is new update data information, it is sent to the data information manager. Otherwise, it is sent to the query processor
data_information_manager	The node is to manage the metadata of the assistant model
update_hold	The queue is to hold new update data information if the update_serv is busy
update_serv	The node is to modify the metadata of the assistant model.
storage_alloc	The node is to allocate spaces in the data information storage to store the new data
storage_alloc2	The node is to allocate spaces in the data information storage to store the new update
space_manager	Full check
change_flow	If the data can't be found in the assistant then go to the data warehouse model. Otherwise, query processor processes the query data
Reference to q_processor	Send the query to the q_processor for processing

Table 4. Parameters used in the simulation model

Parameters	Meanin
job[jobnum].arrvl_time	Data arrival time from the data source to the data warehouse.
quer[s].arrvl_time	User query arrival time from the query source to the data warehouse.
Completion_time	The time a job is finished.
job[jobnum].resp_time	The response time of the data update
quer[s].resp_time	The response time of the query

Table 5. The parameters of the data warehouse

Parameters	Meanin
Data_issue_time	The interarrival time of the external data source
Query_issue_time	The interarrival time of the user query source

Table 6. Data response time of the data warehouse without the assistant

Run Length / Interarrival Time	50	100	250	500	750	1000
External data expo(1.0) query source expo(4.0)	0.118577	0.094158	0.109033	0.137556	0.157109	0.141799
External data expo(2.0) query source expo(5.0)	0.137893	0.231146	0.342046	0.368979	0.402830	0.380082

Table 7. Query response time of the data warehouse without the assistant

Run Length / Interarrival Time	50	100	250	500	750	1000
External data expo(1.0) query source expo(4.0)	1.883286	1.921569	1.981808	1.918018	1.915810	1.938558
External data expo(2.0) query source expo(5.0)	3.129214	2.896530	2.504848	2.403723	2.186110	2.207509

Table 8. Data response time of the data warehouse with the assistant

Run Length / Interarrival Time	50	100	250	500	750	1000
External data expo(1.0) query source expo(4.0)	0.015646	0.024207	0.054216	0.052733	0.060664	0.058016
External data expo(2.0) query source expo(5.0)	0.151044	0.141795	0.160236	0.187258	0.146457	0.156529

Table 9. Query response time of the data warehouse with the assistant

Run Length / Interarrival Time	50	100	250	500	750	1000
External data expo(1.0) query source expo(4.0)	1.626377	1.602235	1.822521	1.873358	1.909446	1.891572
External data expo(2.0) query source expo(5.0)	2.016015	2.362016	2.126121	2.195055	2.244791	2.299350

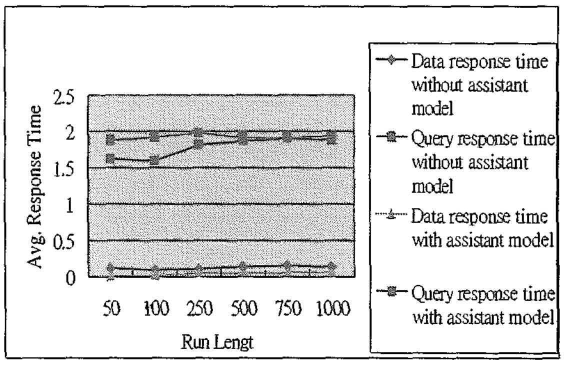

Figure 9. The result with external data expo(1.0) and quer source expo(4.0)

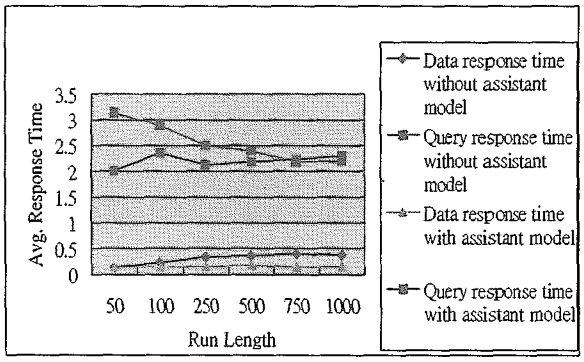

Figure 10. The result with external data expo(2.0) and query source expo(5.0)