

# **Improvement of Data Quality with Timeliness in Information Manufacturing System (IMS)**

**Mohammad Shamsul Islam**

A dissertation submitted in fulfilment of the requirements for the award of  
M.Eng.

Supervisors

Dr. Markus Helfert  
&  
Dr. Paul Young



Faculty of Engineering & Computing  
Dublin City University

May 2014



## **Declaration**

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Master of Engineering is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Shamsul Islam (Candidate) ID No.: 54180309 Date: 10/10/2014



*To My Parents*



## **Acknowledgements**

First of all, I would like to thank my supervisors Dr. Markus Helfert and Dr. Paul Young for their assistance and invaluable advice.

I would like to give special thanks to my parents for their blessing, support, guidance and encouragement. Thanks to my younger sister Sinthiya Islam, elder sister Yesmin Begum and brother in law Nasir Uddin Mozumder for their consistent support. I also want to mention my sweet little nephew Yash Mozumder and niece Nafiza Nuzhat for their indirect support to me.

Finally, I would like to give a big „thank you“ to all members of the Mechanical & Manufacturing Engineering department and especially the ex-chairman, Professor M. S. J. Hashmi, for his help throughout this research.



## Table of Contents

1	Introduction.....	1
1.1	Concept of Information Manufacturing System (IMS).....	2
1.2	Data Quality Issue in Information Manufacturing System .....	4
1.3	Research Questions and Guideline of the Execution .....	6
1.4	Thesis Organization .....	12
2	Literature Review.....	14
2.1	Introduction.....	14
2.2	Information Manufacturing System .....	14
2.2.1	Structural Elements of IMS.....	15
2.2.2	Operational Elements of IMS.....	18
2.3	Pertaining Data Quality Dimensions for Certificating the Quality of Data in IMS .....	21
2.3.1	Most Usable Data Quality Dimensions.....	21
2.3.2	Objective Data Quality Dimensions.....	22
2.3.3	Time Related Data Quality Dimensions.....	25
2.4	Related Research on Data Quality Assessment for IMS .....	28
2.4.1	Assessment Point .....	28
2.4.2	Assessment Class .....	29
2.4.3	Assessment Criterion .....	30
2.4.4	Assessment Method .....	31
2.4.5	Assessment Matrices.....	33
2.5	Data Quality Problem in IMS.....	35
2.6	Data Quality Problem for Updating Data in IMS.....	37
2.7	Time Related Quality Aware Query.....	39
2.8	Trade-off between Timeliness and other Objective Data Quality Dimensions.....	41
2.9	Summary .....	45
3	Heterogeneous Information Manufacturing System (IMS) and Data Quality Constraints for IMS .....	47
3.1	Introduction.....	47
3.2	Heterogeneity of the Structural Elements of Information Manufacturing System.....	48
3.2.1	Degree of Integration of Sources .....	49
3.2.2	Types of DSS .....	50
3.3	Heterogeneity of the Operational Elements of IMS.....	60



3.3.1	Machine.....	60
3.3.2	Material .....	61
3.3.3	Refreshment Processing.....	63
3.3.4	Query Processing .....	64
3.4	Data Quality Constraints in IMS.....	65
3.5	Summary .....	67
4	Data Quality Assessment Functions & Procedures.....	70
4.1	Introduction .....	70
4.2	Objective Data Quality Assessment Function.....	71
4.3	Timeliness of Data in IMS .....	78
4.3.1	Volatility of Data in IMS .....	78
4.3.2	Currency of Data in IMS.....	80
4.4	Comparison of Inbound & Outbound Data Quality .....	98
4.4.1	Comparison Approach of Inbound and Outbound Data Quality.....	99
4.5	Summary .....	103
5	Diversification of Data Quality with Timeliness in Heterogeneous IMS .....	104
5.1	Introduction.....	104
5.2	Diversification of Data Quality with Timeliness in IMS .....	106
5.2.1	Diversification of Data Quality in Heterogeneous IMS in Theoretical Perspective .....	110
5.2.2	Measurement of Data Quality in Heterogeneous Simulated Information Manufacturing System (IMS) .....	128
5.3	Summary .....	151
6	Discussion .....	152
6.1	Introduction.....	152
6.2	Data Quality Discussion for the Theoretical Analysis of Heterogeneous IMS.....	152
6.3	Data Quality Discussion for the Experimental Results of Heterogeneous Simulated IMS .....	154
6.3.1	Data Quality Comparison of DSS Oriented Heterogeneous IMS .....	155
6.3.2	Data Quality with Timeliness for the Execution Method of Refreshment Function in IMS .....	156
6.3.3	Data Quality with Timeliness for the Execution of Number of Tasks for the Refreshment Function of IMS .....	157
6.3.4	Data Quality with Timeliness for the Machine Capacity in IMS.....	159
6.3.5	Data Quality with Timeliness for Refreshment Frequency Method in IMS .....	160
6.3.6	Data Quality with Timeliness for the Overhead Task of Refreshment Function in IMS.....	161
6.3.7	Data Quality with Timeliness for the Volume of Data in IMS .....	162
6.3.8	Data Quality with Timeliness for the Change Frequency of Data in IMS .....	163



6.4	Summary .....	164
7	Conclusion .....	166
7.1	Research Findings .....	166
7.2	Limitations of This Research .....	166
7.3	Future Work .....	167



## **Abstract**

Nowadays in the digital world, organizations or enterprises like banks, hospitals, telecommunications or retail shops etc. has an information manufacturing system (IMS) for storing the organization's data in digital format. Every day, a large quantity of data is manipulated (inserted, deleted and updated) to the information manufacturing system of those enterprises or organizations.

To be successful, the IMS must maintain the data and transform it into useful information for decision makers or users. Much of the value will rest in the quality of the data, which may be divided into two classes; objective and time related. In seeking to maintain quality both these classes the completeness, accuracy and consistency of the data and the timeliness of the information generation may be required. As a further complication, Objective data quality class may not be independent. It could be dependent on timeliness of time related class.

The main purpose of this research is the improvement of data quality with timeliness in IMS. This starts with observing the reasons for the change of objective data quality over time by using both theoretical and experimental data quality measurements. Novel approaches to ensuring the best possible information quality is developed and evaluated by observing the change of objective data quality scenario with timeliness in a purpose built IMS.



## List of Figures

Figure 1.1: Pertaining Data Quality Regulating Factors, Constraints & Dimensions.....	8
Figure 1.2: Data Quality Affecting Process with Timeliness in Heterogeneous IMS.....	10
Figure 1.3: Total Data Quality Management (TDQM) Cycle.....	11
Figure 2.1: Construction of Information Manufacturing System.....	15
Figure 2.2: Structure of Information Manufacturing System.....	16
Figure 2.3: Data Storage System (DSS).....	17
Figure 3.1: Factors for Classifying Heterogeneous Information Manufacturing System .....	47
Figure 3.2: Lowest Degree Integrated Information Manufacturing System .....	49
Figure 3.3: Highest Degree Integrated Information Manufacturing System.....	50
Figure 3.4: Rotation of the Tasks of Functionalities in 3-Data Storage System .....	54
Figure 3.5: Regulator Algorithm for 3-DSS.....	56
Figure 3.6: Synchronizing Agent Algorithm for 3-DSS .....	57
Figure 3.7: Affected Data Quality Dimensions for not Fulfilling Data Quality Constraints .....	66
Figure 3.8: Determinants of Heterogeneity of IMS .....	69
Figure 4.1: Currency of the Data in Single DSS Oriented IMS .....	89
Figure 4.2: Currency of the Data in Cluster DSS Oriented IMS.....	91
Figure 4.3: Currency of the Data in 2-DSS (Temporary) Oriented IMS.....	94
Figure 4.4: Currency of the Data in 2-DSS (Permanent) Oriented IMS .....	95
Figure 4.5: Currency of the Data in 3-DSS Oriented IMS .....	98
Figure 4.6: Random Horizontal Data Selection Approach.....	100
Figure 4.7: Random Vertical Data Selection Approach.....	101
Figure 4.8: Partially Mixed (H+V) Random Data Selection Approach.....	102
Figure 5.1: Information Manufacturing Process in IMS .....	107
Figure 5.2: Changing Process of Data Quality with Timeliness in IMS.....	109
Figure 5.3: General DQ Measurement Scenario in IMS .....	115
Figure 5.4: DQ Measurement Scenario of Single DSS Oriented IMS .....	118
Figure 5.5: DQ Measurement Scenario of Cluster DSS Oriented IMS.....	121
Figure 5.6: DQ Measurement Scenario of 2-DSS Oriented IMS (Temporary DSS).....	123
Figure 5.7: DQ Measurement Scenario of 2-DSS Oriented IMS (Permanent DSS).....	125
Figure 5.8: DQ Measurement Scenario of 3-DSS Oriented IMS.....	127
Figure 5.9: Data Quality Assessment Tool for IMS.....	131
Figure 5.10: Experimental Setup of Data Quality Assessment of Heterogeneous IMS.....	132
Figure 5.11: Inbound Data Quality Assessment Algorithm.....	133
Figure 5.12: Data Quality Assessment Graph for Inbound Data .....	135
Figure 5.13: Data Quality Assessment Graph for Outbound Data.....	136
Figure 5.14: Outbound Data Quality Assessment Algorithm .....	138
Figure 5.15: Outbound Data Quality with Timeliness Graph for IMS1.1 .....	140
Figure 5.16: Outbound Data Quality with Timeliness Graph for IMS1.2 .....	141
Figure 5.17: Outbound Data Quality with Timeliness Graph for IMS1.3 .....	142
Figure 5.18: Outbound Data Quality with Timeliness Graph for IMS1.4 .....	143
Figure 5.19: Outbound Data Quality with Timeliness Graph for IMS2 .....	145
Figure 5.20: Outbound Data Quality with Timeliness Graph for IMS2 (Consistency DQ Problem) .....	146
Figure 5.21: Outbound Data Quality with Timeliness Graph for IMS3 .....	147
Figure 5.22: Outbound Data Quality with Timeliness Graph for IMS4.1 .....	149



Figure 5.23: Outbound Data Quality with Timeliness Graph for IMS4.2 .....	150
Figure 5.24: Outbound Data Quality with Timeliness Graph for IMS4.3 .....	151
Figure 6.1: Data Quality Comparison of DSS Oriented Heterogeneous IMS.....	156
Figure 6.2: Comparison of Data Quality with Timeliness for the Execution Method of Refreshment Function in IMS .....	157
Figure 6.3: Comparison of Data Quality with Timeliness for the Execution of Number of Tasks for Refreshment Function in IMS .....	158
Figure 6.4: Comparison of Data Quality with Timeliness for the Machine Capacity in IMS .....	159
Figure 6.5: Comparison of Data Quality with Timeliness for the Refreshment Frequency Method in IMS	160
Figure 6.6: Comparison of Data Quality with Timeliness for the Overhead Task of Refreshment Function in IMS.....	161
Figure 6.7: Comparison of Data Quality with Timeliness for the Volume of Data in IMS.....	162
Figure 6.8: Comparison of Data Quality with Timeliness for the Change Frequency of Data in IMS.....	163
Figure 6.9: Scenario of Data Quality Variation & Improved Quality Data in IMS .....	165



## List of Tables

Table 1.1: Concept of Information Manufacturing Process in IMS (Cochinwala et al., 2001) .....	3
Table 2.1: Most Usable Data Quality Dimensions.....	22
Table 2.2: Required Elements for Data Quality Assessment .....	28
Table 2.3: Comparison of Objective and Process Assessment of Data.....	30
Table 2.4: Assessment Criterion of Data Quality Assessment Classes.....	31
Table 2.5: Assessment Unit & Ranges of Assessment Classes.....	31
Table 2.6: Requirements of Precise Assessment Score for Assessment Criterion Classes.....	32
Table 2.7: Practicality Requirements of Data Quality Assessment for Assessment Criterion Classes.....	32
Table 2.8: Techniques of Data Quality Assessment for Assessment Criterion Classes.....	33
Table 2.9: Types of Ratings Form for Data Quality Checking .....	34
Table 3.1: Dimensions for the Factors of the Heterogeneous DSS.....	59
Table 4.1: Example of Consistency Problem.....	77
Table 4.2: Currency Parameters of IMS.....	81
Table 4.3: Refreshment Processing Period Parameters.....	81
Table 4.4: Probable Simultaneous Manipulation Operations in IMS from Multiple Sources .....	83
Table 5.1: Heterogeneous IMS for Data Quality Analysis in Theoretical Aspect .....	111
Table 5.2: Notations of Variables and Terms for Measuring the Data Quality with Timeliness in Heterogeneous IMS.....	112
Table 5.3: Data Quality Scenario for Timeliness Factors in IMS .....	114
Table 5.4: Heterogeneous IMS for the Data Quality Measurement in Experimental Aspect .....	130
Table 5.5: Inbound Data Quality Assessment.....	134
Table 5.6: Outbound Data Quality Assessment .....	136
Table 5.7: Experiment Number of Single DSS Oriented Heterogeneous IMS .....	139
Table 5.8: Outbound Data Quality with Timeliness for IMS1.1 .....	140
Table 5.9: Outbound Data Quality with Timeliness for IMS1.2.....	141
Table 5.10: Outbound Data Quality with Timeliness for IMS1.3 .....	142
Table 5.11: Outbound Data Quality with Timeliness for IMS1.4.....	143
Table 5.12: Outbound Data Quality with Timeliness for IMS2.....	144
Table 5.13: Outbound Data Quality with Timeliness for IMS2 (Consistency DQ Problem) .....	145
Table 5.14: Outbound Data Quality with Timeliness for IMS3.....	147
Table 5.15: Experiment Number of 3-DSS Oriented Heterogeneous IMS .....	148
Table 5.16: Outbound Data Quality with Timeliness for IMS4.1 .....	149
Table 5.17: Outbound Data Quality with Timeliness for IMS4.2 .....	150
Table 5.18: Outbound Data Quality with Timeliness for IMS4.3 .....	151
Table 6.1: DQ Comparison of Heterogeneous DSS Oriented IMS in Theoretical Aspect .....	152
Table 6.2: Data Quality Scenario of Heterogeneous DSS Oriented IMS .....	156
Table 6.3: Data Quality Scenario for the Execution Method of Refreshment Function in IMS .....	157
Table 6.4: Data Quality Scenario for the Execution of Number of Tasks for Refreshment Function in Heterogeneous IMS.....	158
Table 6.5: Data Quality Scenario for the Machine Capacity in Heterogeneous IMS .....	159
Table 6.6: Data Quality Scenario for the Refreshment Frequency Method in IMS.....	160
Table 6.7: Data Quality Scenario for the Overhead Task of Refreshment Function in IMS .....	162
Table 6.8: Data Quality Scenario for the Volume of Data in IMS.....	163
Table 6.9: Data Quality Scenario for the Change Frequency of Data in IMS.....	164



# 1 Introduction

---

Currently, in this integrated digital world, the flow of information is increasing. Many businesses have been globalized and therefore, are facing more competition than ever before. The reputation of an enterprise or organization depends on the user's or customer's satisfaction of that organization or enterprise. Poor data quality (DQ) negatively impacts the customer satisfaction at the operational level and may result in lost customers.

Incorrect information is often cited as the root cause of massive economic losses suffered by enterprises or organizations. More than \$2 billion U.S federal loan money has been lost due of poor DQ in a single agency (Wang et al., 1995). One major financial institution was embarrassed because of an incorrect data entry of an order of \$500 (Wang et al., 2001). Another telecommunication company lost \$3 million due to poor information quality (IQ) in customer bills (Huang et al., 1999). It is estimated that poor IQ results in 8% to 12% loss of revenue. This is typical in enterprises, and is estimated to be responsible for 40% to 60% of expense in service organizations (Redman, 1998).

Poor quality data can affect decision quality. The NASA's space shuttle challenger accident in the 28<sup>th</sup> January 1986 and the Iranian commercial aircraft that was shot down on July 1988 are prime examples of poor quality decisions for harmful disaster which are all due to poor IQ (Fisher and Kingma, 2001). One explanation for poor IQ can be traced to the poor quality of the data that underpins it.



Enterprises and organizations need to deliver information for decision making in both real time and non-real time environments. Furthermore, many enterprises operate on a  $24 \times 7$  business time schedule. Stock brokering, e-business, online telecommunication, healthcare system and traffic system need to deliver information at high-speed to decision makers who make decisions in a real time environment (Inmon et al., 2001). This delivered information is manufactured by the information system of the enterprises or organizations. Failure to send the mandatory information to the right user at the right moment leads to poor DQ. Poor DQ at the decision level can affect decision quality. A single decision is often based on the evaluation of large amounts of data. Detection of the poor DQ may take a great deal of time and often the required data is not available. Therefore, the decision-making process can be affected from lack of essential information.

### **1.1 Concept of Information Manufacturing System (IMS)**

The information manufacturing system can be compared to a product manufacturing system. An information manufacturing system manufactures information from raw data. Similarly, the product manufacturing system manufactures products from raw materials. Both the information manufacturing system and the product manufacturing system have inputs, processes and outputs. Therefore, there is no conceptual difference between the information manufacturing system and the product manufacturing system for manufacturing information and products respectively. This is shown in Table 1.1.



**Table 1.1: Concept of Information Manufacturing Process in IMS (Cochinwala et al., 2001)**

Stage	Product Manufacturing	Information Manufacturing
Input	Raw Materials	Raw data
Process	Assembly Line	Information System
Output	Physical Product	Information Product

However, the information manufacturing system differs from the product manufacturing system by the input of manufacturing elements. Raw materials for product manufacturing are consumed when manufacturing the physical product. In contrast, raw materials used in information manufacturing, although used, are not consumed. In the digital manufacturing (DM) industry, large amount of data and information are collected during the product manufacturing process. The manufacturing process of each type of information begins from the information source, together with gathering, processing, storing and transforming before finally reaching the information consumer.

Nowadays, in the digital world, organizations and enterprises like banks, hospitals, telecommunications and retail shops, must have an information system for information support. On a daily basis, a large volume of data is managed in the information system of the enterprises or organizations for information support of the organization or enterprise in both real time and non-real time environments. Three types of data such as raw data, component data and information product are associated with the information system (Batini and Scannapieco, 2006). Scattered raw data come from multiple data sources. These are integrated and transformed into aggregated raw data in the data storage system. Therefore, aggregated raw data turns into component data or information products for the demand of customers, or organizations. In this sense,



organization or enterprise information systems can be recognised as the information manufacturing system (Wang et al., 2001).

## **1.2 Data Quality Issue in Information Manufacturing System**

The information product of the information manufacturing system may be poor quality. Inconsistencies in data, missed or imprecise values and unacceptable performance are the components of poor DQ. DQ problems of multiple source data can occur at the time of data entry (source level), in addition to the time of integration (transformation). DQ problems at the source level and data integration level can be solved by DQ products in the market.

Many authors define DQ as “fitness for use”. This goes beyond just considering the accuracy of the data. Rather, it sees DQ as a multidimensional and context dependent concept. Various authors propose different DQ dimensions. An analysis of these dimensions is given in detail in chapter 2. Timeliness is identified as the most useable time related DQ dimension. The most useable objective DQ dimensions were identified as accuracy, completeness and consistency. These specific DQ dimensions are used in information (manufacturing) system such as cooperative information system, web information system, and multi-channel information system for assessing DQ.

DQ dimensions are used to measure the quality of data. Objective DQ dimensions are used for assessing and measuring the DQ of stable or long-term changing data in IMS. Time related DQ dimension may be used for measuring and assessing long-term changing data. For an example, if a volume of data needs to be inserted in IMS where the expiry date of those data is „x“ or more days, timeliness of these data is the timeliness of the long term changing data. Timeliness of these data is high for the long



expiry time. In this case, if data need more time (i.e. an hour or more) to insert in IMS, these data will not be obsolete. In this instance, the timeliness dimension of data is less important than ensuring other DQ dimensions such as completeness, consistency etc. However, sometimes, both timeliness and objective DQ dimensions are of equal importance for good IQ. For example, someone needs information about a location that they wish pass through within „x“ minutes and so sends a request to the IMS to receive that information. If the IMS does not deliver that information within the required „x“ minutes, but delivers it after the required „x“ minutes, it will be of little or no value even if it is complete and accurate and it will be deemed to be poor IQ for the dimension of timeliness. However, if the information arrived on time but was inaccurate or incomplete, the information would also be deemed as poor IQ.

Customer service departments of an organization, i.e. supermarket (wall mart), or banks, needs to refresh data in the IMS for long term and frequently changing data either in non-real time or in a nearly real time manner. The most important concern is to deliver complete, consistent and accurate data or information on time. Moreover, data changes frequently in the IMS of air traffic, road traffic, e-business, stock-brokering and air ticket reservation system data should be updated without delay to provide vital up to date information, For example, an air traffic control system monitors hundreds of aircrafts and makes crucial decisions about incoming flight paths, determines the landing order of aircrafts based on data such as fuel level, altitude and speed. If any of this vital information is late, missing or inaccurate the result could have devastating consequences.

Occasionally, there are trade-offs between timeliness and the DQ dimensions of accuracy, completeness and consistency (Batini and Scannapieco, 2006). Basically,



these trade-offs are important considerations for time related data and real time data. If a query request is sent to the IMS data storage system of while manipulated data are being refreshed and there is an obligation to respond to the query request within a certain time frame, the request may respond with an incomplete and inaccurate data set to comply with the obligation of timeliness. Undeniably, having timely data may cause lower accuracy, completeness or consistency. Conversely, to have accurate, complete and consistent data, timeliness may be negatively affected (Batini and Scannapieco, 2006). Furthermore, there is a time related accuracy and completeness problems in IMS (Cappiello et al., 2003). Hence, if timeliness is negatively affected, objective DQ dimensions are not affected and if, timeliness is not affected, objective DQ dimensions are affected.

### **1.3 Research Questions and Guideline of the Execution**

When analysing DQ issues in IMS, it is found that DQ problems may occur in the IMS used in providing information support in both real time and non-real time environments. A DQ problem occurs for timeliness and objective DQ dimensions, such as, accuracy, completeness and consistency. Moreover, there is a relationship between timeliness and objective DQ dimensions and therefore, the timeliness dimension can make an impact on objective DQ dimensions. Hence, this research will be guided by the following research questions.

Q1: How can timeliness change DQ in IMS?

Q2: Is it possible to obtain improved quality data with timeliness in IMS?

Timeliness issues of DQ can cause poor DQ in IMS. Timeliness DQ factors such as age, delivery time, input time and volatility plays the role in changing DQ.



Furthermore, these factors may be influenced by other factors involved in the information manufacturing system. Thus, the purpose of the first research question is to identify those factors for which timeliness changed DQ in IMS.

It is possible to improve data or IQ by changing DQ regulating factors in the IMS. Hence, the target of our second research question is to identify a set of factors that can improve data quality with timeliness.

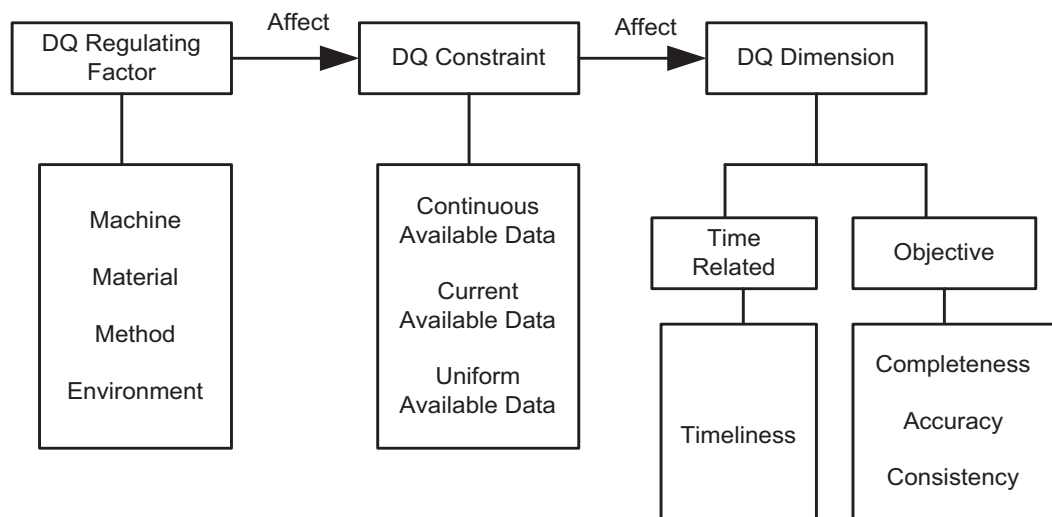
According to Wang et al. (2008) data or IQ can be influenced by machine, material, methods and environment.

- Machine in the IMS involves the data processing unit. This data processing unit is the combination of hardware and software (database) of the IMS.
- Material is the raw data of the IMS. Raw data can be categorized as new raw data and existing raw data in the IMS.
- In IMS, method is the way or technique of processing and delivering information (Wang et al., 2008).
- Environment can be identified as the operational environment of the IMS. In this sense, operational environment can be divided into real time and non-real time environment.

DQ constraints in the IMS are categorized as uniform available data, continuous available data and current available data (Capiello et al., 2005; Capiello and Helfert, 2008). No mismatching the simultaneously delivered data from IMS to the users is defined as the uniform available data. Service of required data delivery from IMS without interruption can be named as continuous available data. Furthermore, delivery of the most recent data that is not obsolete from IMS is called current available data.



Both, time related DQ dimension and non-time related objective DQ dimensions are discussed in more details in chapter 2. Timeliness DQ dimension is considered for data timeliness and information (a set of data) timeliness. Information timeliness can be defined as whether or not outbound data comes from the system at the right time. Alternatively, data timeliness can be defined by the ratio of currency and volatility of the data. DQ regulating factors affect the DQ constraints. DQ dimensions are subsequently affected by these DQ constraints as shown in Figure 1.1.



**Figure 1.1: Pertaining Data Quality Regulating Factors, Constraints & Dimensions**

According to Capiello and Helfert (2008) there is a trade-off between availability and timeliness DQ dimensions for refreshment frequency in the IMS. Availability dimension makes an impact on the completeness DQ dimension. Incompleteness of data will occur for the non-continuity of available data. On the other hand, continuous available data can ensure completeness of data. Complete data can be of poor quality if not current and uniform. Accuracy and consistent DQ dimensions can be ensured by current and uniform available DQ constraints in the information manufacturing system. So, accuracy and consistency DQ problems can occur for currency and uniformity DQ

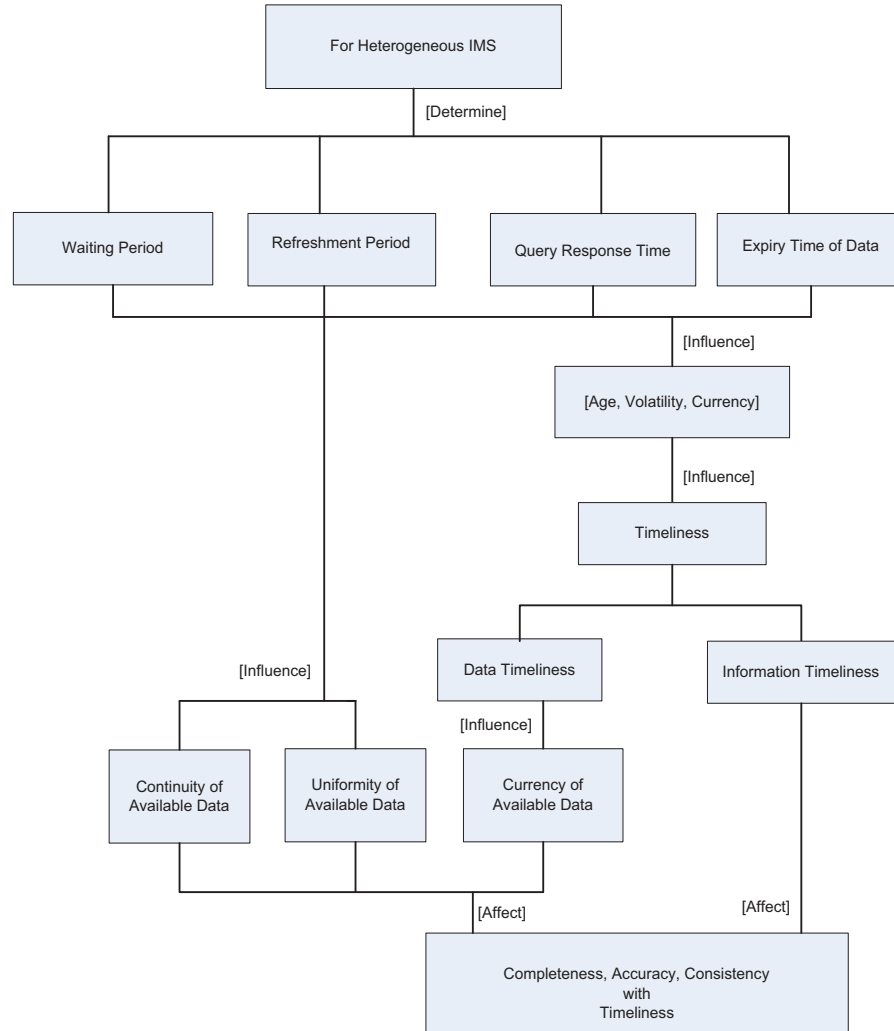


factors in IMS. As a result, objective DQ dimensions can be affected by the time related or timeliness DQ dimension in the information manufacturing system.

Effect on DQ by timeliness depends on the degree of co-ordination between the outbound data demands and the inbound data supply. Continuous available data in the IMS can ensure the high degrees of co-ordination between the actual outbound data demands with inbound data supply. In Figure 1.2, DQ affecting process with timeliness of IMS is shown. Degree of co-ordination between the actual demands of outbound data with inbound data supply may vary with the refreshment processing mechanism. The refreshment processing mechanism varies for heterogeneous IMS. This refreshment processing mechanism includes refreshment frequency and refreshment period. Refreshment frequency (continuous or periodic) and other overhead times can have an effect on the waiting period. Therefore, the refreshment period and the waiting period will be determined for heterogeneous IMS. The refreshment period influences the continuity and uniformity of available data. Expiry time of data varies for the change frequency of data. Query response time is the responding time for the query request. So, expiry time of the data and the query response time are determined for the heterogeneous IMS. Both data timeliness and information timeliness are discussed later in this thesis. These two timeliness functions are influenced by the waiting period, refreshment period; query response time and expiry time of data shown in Figure 1.2. Currency of available data is motivated by the data timeliness value of each individual data. The degree of co-ordination between outbound data demand and inbound data supply is regulated by information timeliness and DQ factors such as continuity of available data, current available data and uniformity of available data. Therefore,



information timeliness affects the completeness, accuracy and consistency DQ dimensions in IMS.

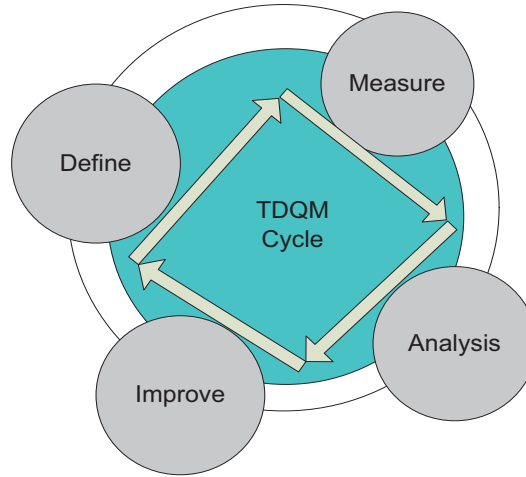


**Figure 1.2: Data Quality Affecting Process with Timeliness in Heterogeneous IMS**

Heterogeneous IMS is used to provide answers to the research questions. Heterogeneous IMS is discussed in detail in chapter 3. The heterogeneous IMS is used for the theoretical analysis and simulation. Theoretical analysis and experimental results of the simulated heterogeneous IMS will show how timeliness changes DQ. The reasons of the diversification and the improvement of DQ by timeliness will be found from theoretical analysis and experimental results.



Many DQ researches follow four phases of the total DQ management model (Wang, 1998). This research will also follow these four phases.



**Figure 1.3: Total Data Quality Management (TDQM) Cycle**

Define is the first phases of the total DQ management system. The define component of the TDQM cycle identifies important DQ dimensions and the corresponding DQ requirements. The main task of DQ research is to define each individual DQ dimension by considering it in the context of research. A unique and clear definition of DQ is the core for any assessment or improvement approach.

Measurement or assessment phase 2 produces DQ assessment matrices or functions. The assessment phase can rely on questionnaires in a subjective point of view. The techniques measure is based on the association of a question with each considered dimension and the inference of the metric from the answer given by the interviewee. Assessment can also be done from an objective DQ point of view. Automated evaluation technique is used for measuring the DQ of the system's data with the assessment functions.



The analysis phase 3 identifies the root causes of DQ problems. Analysis is done by assessment of the result measured by DQ dimensions.

Finally, DQ problems are identified and, the improvement phase 4 provides a possible solution for improving DQ.

This research shows the change of data quality with timeliness in IMS. Therefore, improvement of data quality is shown from these changes. In the real world, data comes from source to DSS in IMS. At the same time query can be executed in IMS. Different DSS oriented IMS can be used in the real world. Therefore, heterogeneous IMSs are simulated in this research. The query function is executed at the time of refreshment function execution period. Therefore, the query results are assessed by the assessment function. Then, changes of DQ are found for the IMSs. Finally, improvement of DQ is found by comparing the DQ of heterogeneous IMSs.

## **1.4 Thesis Organization**

This thesis is organized as follows: chapter 2 provides a review of the relevant DQ and information manufacturing system literatures. Chapter 3 presents heterogeneous IMS and the constraints of DQ for IMS. Chapter 4 outlines the assessment functions and procedures for assessing inbound and outbound data of simulated IMS. Chapter 5 details DQ measurements for the timeliness dimension of DQ in heterogeneous IMS. Chapter 6 provides a detailed discussion of the research findings. Chapter 7 concludes the thesis by summarising the research findings and outlining some future research possibilities. The specific organization of each chapter is as follows:

In chapter 2, a review of the DQ literatures on information manufacturing system is presented. Structural and operational elements of the information manufacturing system



are discussed for devising a complete IMS for manufacturing information. The most usable DQ dimensions in IMS as defined by researchers are discussed for good quality manufactured information. Some assessment elements are also discussed. DQ problems in the IMS of the real world are also examined. DQ problems at the point of updating data are reviewed. Time related quality aware query is used in IMS for the quality purpose of data. These queries are further discussed in chapter 2. Finally, a discussion of the trade-offs between time related and objective DQ dimensions are reviewed.

Heterogeneous IMS and DQ constraints for IMS are elaborately discussed in chapter 3.

Data assessment functions for both objective and timeliness DQ dimensions are discussed in chapter 4. This chapter also shows the procedure of assessment for measuring DQ in both inbound and outbound data of the IMS.

Poor DQ and DQ improvement in IMS by timeliness are revealed in chapter 5, in both the theoretical (math modelling) and an experimental aspect. Experimental tool, setup for getting experimental results for heterogeneous simulated information manufacturing system is also discussed. The experimental results of heterogeneous simulated IMS are provided to show improvements of DQ in the IMS.

Chapter 6 provides an overall discussion of this thesis. DQ of heterogeneous IMS is analysed and compared in a theoretical aspect. In addition, the experimental results of heterogeneous IMS are compared and the research findings shown.

Chapter 7 provides a critical review of the study and the limitations of this work are discussed. Finally, an indication for future research is put forward.



## **2 Literature Review**

---

### **2.1 Introduction**

Data or IQ is the universal term in the technology and business field of research. Much research on DQ has been conducted in these two fields. The technological field covers information technology, and telecommunication etc. In contrast, the business field works on the management information system (MIS). Both the technology and business fields tie up with the information system (IS). But it varies from the level to level (from workers to executive information system), and department to department etc., in an organization or a company. The main reason for setting up an IS is to manufacture information. Managerial parts (CRM, SCM) work under the IS for manufacturing information. The managerial part of the IS is called MIS. Therefore, data or IQ researches are either in IS centric or IMS centric. In this chapter, IMS and the relevant DQ research on IMS is discussed for establishing the motivation of this research.

### **2.2 Information Manufacturing System**

The IMS is the system that manufactures information from raw data (Wang et al., 2001). This IMS is made-up by the structural and operational elements of the IMS. The structural elements of the IMS form the structure of the IMS. The operational elements make the structure of the IMS operational for manufacturing information. The IMS is depicted in Figure 2.1.



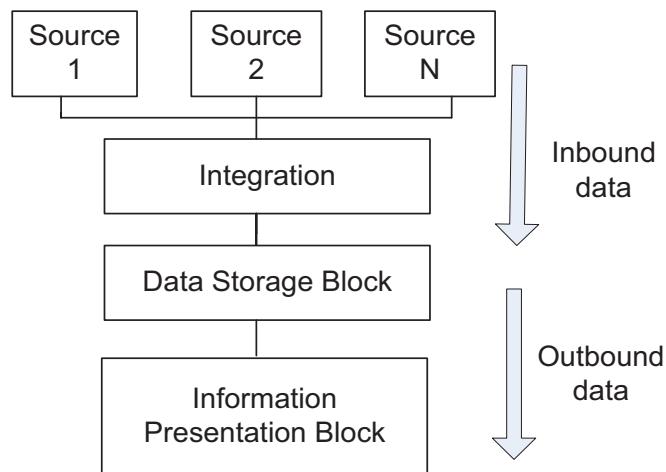
Information Manufacturing System	Structural Elements	Elements
		Source
		Integration
		Data Storage System
		Information Presentation
	Operational Elements	Machine
		Material
		Refreshment Processing
		Query Processing

**Figure 2.1: Construction of Information Manufacturing System**

### **2.2.1 Structural Elements of IMS**

The structural elements of the IMS are source block, data storage block and information presentation block (Ballou et al., 1998). The source block, may have multiple sources for storing data in the data storage block (Capiello et al., 2005). IMS is formed by the source block, integration mechanism, data storage block and information presentation block (Ballou et al., 1998; Capiello et al., 2005). The formation of the structure of the IMS is shown in Figure 2.2.





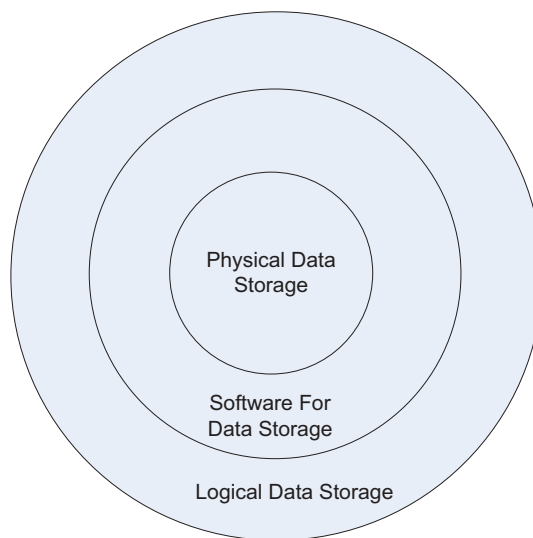
**Figure 2.2: Structure of Information Manufacturing System**

**Source Block:** Data source block represents various sources of raw (input) data. The IMS may have multiple sources in one single place or in multiple different places. Source block does not have a predecessor block. Thus, source block can potentially supply several different types of raw data. Furthermore, the business role responsible for the raw input data, (the data elements that make up the raw data and underlying system in which it is stored and from which it originates) are associated with this block (Ballou et al., 1998). Sources of the IMS could be „a form“, „an automatic sensor system“ or „an operational database“ (ODS). The „form“ source and „automatic sensor system“ can store data in the operational database data storage system. Alternatively, the operational database source data can be extracted and stored in the data warehouse (DW) or the data mart data storage system.



**Integration of Sources:** Sources are integrated to store the extracted or inserted data from multiple sources, multiple functionalities or multiple channels, in the data storage system. Integration of the sources is established in different degrees. The degree of integration of sources differs for the cost (Capiello et al., 2005).

**Data Storage Block:** Data storage block models the placement of data unit in storage files or databases so that they are available for further processing. Storage blocks may be used to present data items (raw and/or component) that wait for further processing or captured as part of the information inventory in the organization (Ballou et al., 1998). The data storage block can be constructed by some data storage systems. This data storage system is formulated by the technology like operational database system, data warehouse and data mart etc., depending on the size of the organization and the frequency of transactions (Ha and Park, 1998). In these technologies, the logical structure of the data storage system is not different. The construction of a data storage system is depicted in Figure 2.3.



**Figure 2.3: Data Storage System (DSS)**



The data storage system is constructed with the physical data storage, the software of the data storage, and the logical data storage system (Rizvi and Chung, 2010; Santos et al., 2008). The physical data storage system is where the software of the data storage system is installed. Flash memory and hard drives are examples of physical data storage systems. Oracle, SQL SERVER, MYSQL, and IBM DB2 are examples of the software for the data storage system (Rizvi and Chung, 2010). A logical data storage system is created in the software of the database data storage system. A table is a logical data storage system for storing data that comes from multiple sources. It is the combination of a set of attributes and tuples. An attribute is a single data item related to a database object. The tuple is the set of data in a row. The record is the individual attribute data of a tuple (Silberchatz et al., 1997). Records are indexed in the table for the improvement of performance of information support (Santos et al., 2008). Indexing is discussed further in section 2.2.2.

**Information Presentation Block:** This block represents the output or information product of the IMS. The name of the entity that actually uses the product and the set of data items that make up the information product are associated with this block (Ballou et al., 1998).

### **2.2.2 Operational Elements of IMS**

Machine, material and the processing mechanism of data, play a role for manufacturing information in the IMS. Software of the structural elements needs to install into the machine. The materials then need to flow into the machine for manufacturing information by refreshment and query processing. Therefore, the presence of the machine, material and the processing mechanism make the IMS operational. This operational IMS is represented as the IMS in this thesis.



**Machine:** Software of the structural elements of the IMS are installed in the machine. Machine in an organizational context, means the data processing unit (Wang et al., 2008). Both hardware and software have a processing unit or function (Zhou and Ding, 2006). The hardware of the machine includes the processor, memory and data storage disk. Therefore, the machine is the combination of hardware and software (Rizvi and Chung, 2010; Santos et al., 2008; Zhou and Ding, 2006).

**Material:** Raw material for the manufacturing information product in the IMS is the data generated from the sources. The source data is stored in the data storage system. The stored data is then presented in the presentation block for information support.

**Processing Block:** Processing block processes the raw or component data for producing an information product in the IMS (Ballou et al., 1998). Data that comes from the source to the data storage block is represented as inbound data. Data that comes from the data storage block to the information presentation block is represented as outbound data. The processing block works on processing both inbound and outbound data. Therefore, this block can be represented as inbound data processing block and outbound data processing block. Refreshment and query function execute in the inbound data processing block and outbound data processing block respectively. Refreshment is the process that helps to make the inbound data available in the data storage system (Mannino et al., 2006). The query is the process of accessing information. Outbound data or information is retrieved from the data storage system by query processing (Nauman and Rolker, 2000).



According to Santos et al. (2008); Bouzeghoub et al. (1999) and; Mannino et al. (2006) the refreshment and query function execute in the data storage system of the IMS to make data available and for the information support respectively.

**Refreshment Function:** Comprising tasks, such as data loading, indexing and propagation of data for synchronizing the data in the information manufacturing system is a complex process (Santos et al., 2008; Bouzeghoub et al., 1999; Mannino et al., 2006).

**Data Loading:** Storage of manipulated (insert, update) data is extracted from the sources and transformed. If source data are in different formats DQ checking may be required. After that, extracted and transformed data are integrated for loading in the data storage system (Santos et al., 2008; Bouzeghoub et al., 1999; Mannino et al., 2006).

**Indexing:** Index is to update for newly loaded data or deleted data in order to align the data in the data storage system (Santos et al., 2008). Indexing determines the effective usability of data collection, aggregation of data from the sources and increases the performance of the data storage system for information support (Bouzeghoub et al., 1999; Mannino et al., 2006).

**Propagation of Data:** Data are propagated through the refreshment process for synchronizing data of multiple DSS of the system.



**Query Function:** The function of the data storage system in the IMS is performed by the query processing task. The requested query of the user is processed in the data storage system to deliver the required information to the user.

### **2.3 Pertaining Data Quality Dimensions for Certificating the Quality of Data in IMS**

Information product quality is of great importance (Ballou et al., 1998). Shankarnarayanan et al. (2000) presents the DQ block as the block used for checking DQ on data items that are essential in producing a „defect free“ information product. Therefore, a list of DQ checks is associated with this block in order to check DQ of specified component data items. According to Ballou et al. (1998) the quality block enhances DQ so that the output stream has a higher quality level than the input stream.

DQ is a multifaceted concept. Numerous authors define DQ as „fitness for use“, i.e. the ability of a data to meet the user's requirements (Wang, 1998; Orr, 1998). The user requirements for DQ are organized by DQ dimensions.

#### **2.3.1 Most Usable Data Quality Dimensions**

Wand and Wang (1996) proposed 5 DQ dimensions. Redman (1996) proposed 17 DQ dimensions regarding the conceptual view of data, data values and data format. Wang and Strong (1996) selected 15 different dimensions from an empirical point of view. Jarke (1999) proposed 22 DQ dimensions of data warehouse quality. Bovee (2001) and Nauman (2002) proposed 10 and 21 DQ dimensions respectively. Caterci and Scannapieco (2002) gave a DQ dimensions comparison table for authors, which show that accuracy; completeness; consistency and timeliness are proposed by almost all of the researchers.



Ge and Helfert (2008) organized and categorized DQ dimensions for assessment of data in the IMS, and accuracy, completeness, consistency and timeliness are grouped into one category. The dimensions of this category are for the assessment of internal data in the IMS. Wand and Wang (1996) proposed DQ dimensions of accuracy, completeness, consistency, timeliness and reliability for the internal view of the information system. Redman (1996) classifies four DQ dimensions for the data values of the data storage system in the IMS. The four DQ dimensions are accuracy, completeness, consistency and currency. Therefore, accuracy, consistency, completeness and timeliness can be identified as a set as the DQ dimensions of the IMS. Further, in the IMS, Batini and Scannapieco (2006), Capiello et al. (2005) and Ballou and Pazer (1985) considers non-time related and time related DQ dimensions as shown in Table 2.1.

**Table 2.1: Most Usable Data Quality Dimensions**

Relevancy of Time	Dimension	Definition
Non-Time Related	Accuracy	It is defined as the degree with which the stored value is consistent with the part of the real world that it has to represent
	Completeness	It considers if all values associated with a specific variables are stored in the system
	Consistency	The data are consistent if the representation of all data values is the same in all occurrences
Time Related	Currency	It concerns how promptly data are updated
	Volatility	It characterizes the frequency with which data vary in time
	Timeliness	Timeliness expresses how current data are for the task at hand. It is defined by the currency and volatility

### 2.3.2 Objective Data Quality Dimensions

Objective DQ dimensions are usually favoured for assessing or measuring DQ from the experimental point of view. Most usable DQ dimensions used as objective DQ dimensions are completeness, accuracy and consistency.



**Completeness:** Completeness in the relational model can be characterised with respect to the presence/absence of null values. The presence of a null value has the general meaning of „missing value“; i.e. a value that exists in the real world but for some reason is not available. Completeness can be generally defined as to the extent to which data are of sufficient breadth, depth and scope for the task at hand (Batini and Scannapieco, 2006). According to Redman (1996) data completeness is defined as an objective dimension and specifies the degree to which specific values are included in a data collection.

Completeness considers, if, all values associated to a specific variable are stored in the system. If, a data item is not stored, its value is null or incomplete. The completeness of the information manufacturing elements ensures the completeness of the data storage system. For an example: if, the fields of the attributes are complete by records, the data set of the table in the data storage system is complete and therefore, ensures the quality from a completeness point of view. Incompleteness may occur in a data entry fault, a system fault, or by late arrival of data in the data storage system. Incomplete inbound data may propagate to the outbound data through the data storage system and may cause a poor IQ product.

Whether all values associated to a specific attribute are stored in the system or not and the late arrival of data in the data storage system is considered for the completeness problem in this thesis.

**Accuracy:** Wang and Strong (1996) define accuracy as, „the extent to which data are correct, reliable and certified. According to Batini and Scannapieco (2006) two kinds of accuracy can be identified in the data of the DSS, one is syntactic accuracy and the



other is semantic accuracy. Syntactic accuracy is the closeness of a value „v“ to the elements. Syntactic accuracy does not check spelling errors or missing letter errors. Rather, it checks whether the inserted value is one of the correct values of the domain. For an example, if the inserted value should be „John“, but, „Jack“ has been inserted, the value is syntactically correct as Jack is an admissible value in the domain of the person's name. But, in actual sense, syntactically correct value is inaccurate as others values of that row may not be the property of the inserted name. Therefore, the name will be inaccurate or else the others values of that row will be inaccurate. On the other hand, semantic accuracy is the closeness of the value „v“ to the true value of „v“<sup>o</sup>. Therefore, if a wrong value is inserted into the DSS, the value will be semantically inaccurate. When the real world entity represents a direct source of information then it is possible to establish the accuracy of a data value that represents a real world entity value. Data values can be inaccurate for collecting data from multiple different sources as data values are stored in the DSS with a complex transformation process. Further, as data are collected from multiple sources, a data value can be incorrect because it is affected by a syntax error, or by ambiguity in its representation. Moreover, data duplication is a cause for inaccuracy of time related data.

Ballou and Pazer (1985) define accuracy as „the degree to which stored value is consistent with the real world“. According to this definition, if data are inconsistent, there must be at least one inaccurate value.

Non-current data is the cause of inaccuracy of data for time related issue (Capiello et al., 2003). The current available data constraint may cause obsolete data in the IMS. As a result, data stored in the DSS of the IMS may be incorrect for obsolescence.



In this thesis, wrong data and obsolete data for the timeliness DQ dimension in the system are considered for the inaccuracy of data.

**Consistency:** Consistency denotes that two or more values do not conflict with each other. Consistency dimension captures the violation of semantic rules defined over a set of data items, where the items are the data sets of the relational tables. The multi-source information system frequently face consistency problem due to the manipulation process from multiple places. Outbound data cannot perform a good result from the consistency problem caused by the inbound data stored in the data storage system of the IMS.

Consistency is defined as the property of multiple data values that do not conflict with each other (Capiello et al., 2005). If, the DSS of the IMS fails to provide uniform data, there will be a consistency problem with the data. As a result, it will violate the uniformity of the information constraint. This will cause poor quality time related data in the IMS. Whether the IMS is delivered uniform information or not is considered in this thesis for showing the consistent and inconsistent information.

### **2.3.3 Time Related Data Quality Dimensions**

Different types of data are stored in the IMS. Obsolescence of a data in the IMS can be identified by the validity measurement of data. The validity of these data can be measured by the timeliness of data or information. Timeliness of data depends on the currency and volatility factors of the data. Factors of timeliness DQ dimensions and the timeliness itself is now discussed.

**Currency:** According to Redman (1996) currency is the degree to which data is up-to-date. A data value is up-to-date if it is correct in spite of possible discrepancies caused



by time related changes to the correct value. In Ballou et al. (1998) currency is defined as a time point that indicates the time instant when data are stored in the data storage system. Jarke (1999) describes currency as when information was entered into the sources or the DSS. Segev and Weiping (1990) define currency by the gap between the extraction of data from the sources and delivery to the users. For example, currency indicates how „stale“ the account balance is when presented to the user with respect to the real balance at the bank. ~~Redman (1996) also define currency as a measure to which data are up to date.~~ According to Batini and Scannapieco (2006) currency is defined as,  $\text{Currency} = \text{Age} + (\text{Delivery Time} - \text{Input Time})$ , Where Age measures how old the data unit is when received, Delivery Time is the time when information product is delivered to the user and Input Time is the time when data unit is obtained. 🟡

~~Currency definition of different researchers is discussed above. Currency definition of Batini and Scannapieco (2006) can measure the currency of data in IMS. Therefore, currency definition of Batini and Scannapieco (2006) will be used in this research.~~

**Volatility:** Volatility characterizes the frequency with which data vary in time. For instance, stable data such as birth dates have volatility equal to 0, (they do not vary over time). Conversely, stock quotes, a frequently changing data, have a high degree of volatility (they remain valid for very short time intervals). Jarke (1999) describes volatility as the time period for which information is valid in the real world. A metric for volatility is the length of time data remains valid (Batini and Scannapieco, 2006). Redman (1996) states that volatility is the measure of the data instability and related to the frequency with which data changes over time.



It needs to know the validity of data in IMS for timeliness calculation. Therefore, volatility definition of Batini and Scannapieco (2006) will be used in this research.

**Timeliness:** According to Batini and Scannapieco (2006) timeliness expresses how current data are for the task at hand. The timeliness dimension is motivated by the fact that it is possible to have current data that are „useless“ because they are late for specific usage. For instance, timetable for a university courses can be current by containing the most recent data, but not timely, if it is not available till after the start of the classes. For Wang et al. (1993) currency and volatility are combined with an average function to obtain the timeliness value.

Timeliness ranges from 0 to 1 where 0 means poorer timeliness and 1 means better timeliness.

According to Batini and Scannapieco (2006) timeliness implies that data not only are current, but also in time for events that correspond to their usage. Therefore, a measurement consists of a currency measurement and a check that data are available before usage time. In Wang and Strong (1996) timeliness is defined as the extent to which data are timely for use. Ballou et al. (1998) define timeliness as „the property of information to arrive early or at the right time“. Therefore, timeliness of data in the IMS depends on whether data are available in time or not.

Obsolescence of data for the time related DQ dimensions and whether information is coming from the IMS in time or not is considered in this thesis.



## 2.4 Related Research on Data Quality Assessment for IMS

According to Ballou et al. (1998) information products are manufactured by multiple stages of processing and are based on data that have various levels of quality. Therefore, assessment of the quality of the information product is important. Many research papers identify the required elements of assessment for assessing the DQ of the IMS (Nauman and Rolker, 2000; Bobrowski et al., 1999; Wang et al., 2002; Ge and Helfert, 2008).

**Table 2.2: Required Elements for Data Quality Assessment**

Required Assessment Element	Dimension
Assessment Point	<ul style="list-style-type: none"><li>• Information Sources (DSS)</li><li>• Query Processing</li></ul>
Assessment Class	<ul style="list-style-type: none"><li>• Objective Assessment</li><li>• Process Assessment</li></ul>
Assessment Criterion	<ul style="list-style-type: none"><li>• Objective Assessment</li><li>• Process Assessment</li></ul>
Assessment Method	<ul style="list-style-type: none"><li>• Objective Assessment</li><li>• Process Assessment</li></ul>
Assessment Matrices	<ul style="list-style-type: none"><li>• Objective Assessment Matrices</li><li>• Process Assessment Matrices</li></ul>

### 2.4.1 Assessment Point

According to Nauman and Rolker (2000) IQ may be assessed in two individual points of an IMS namely: the source point and the query processing point. Moreover, the IQ on these points is influenced by two main factors: the data of the source and the process of accessing the data.



**Data Source:** It is the place where data is stored for producing information. The data storage system of the IMS is the source of data for manufacturing information. Further, the data source itself is the origin of IQ scores for many criteria. It is discussed in section 2.4.3.

**Query Process:** The process of accessing data is a source for IQ scores. DQ assessment criteria for query processing can be automatically assessed during the query process without input from the user or from the data source.

#### **2.4.2 Assessment Class**

Assessment of the data of two points of the IMS is classified as objective assessment and process assessment.

**Objective Assessment:** The objective assessment procedure is used to assess objective IQ. Software is used to automatically measure the DQ in a data storage system by a set of rules or assessment function. A single assessment result is obtained from the objective assessment.

**Process Assessment:** The assessment procedure of a process assessment is the same as the assessment procedure of an objective assessment. Assessing the storage data of data storage system of the IMS, process assessment, assesses the query result. Assessment is done automatically by the software. Furthermore, assessment results may be single or multiple.

Some comparisons of objective and the process assessment of data or information are given in the Table 2.3.



**Table 2.3: Comparison of Objective and Process Assessment of Data**

<b>Class</b> <b>Feature</b>	<b>Objective Assessment</b>	<b>Process Assessment</b>
Tool	Software	Software
Measuring Object	Stored Data	Query Result
Measuring With	Rules, Function	Function
Process	Automated	Automated
Assessing Result	Single	Single/Multiple
Dependency	Data Storage System (DSS)	Query Processing

### **2.4.3 Assessment Criterion**

Assessment criterion is the DQ dimensions used for assessing DQ in a system. This could be varied or the same for both objective and process DQ assessment.

**Objective Assessment:** It is done by considering the objective IQ criterion. Scores on an objective assessment can be determined by a careful analysis of information. Thus, the source of their scores is the information itself.

**Process Assessment:** It is done by considering the process IQ criterion. Scores of a process assessment are determined by the process of querying. The source of the scores is the actual query process. Scores vary from query to query.

A list of the assessment criterion in two individual classes addressed by Nauman and Rolker (2000) is given in Table 2.4.



**Table 2.4: Assessment Criterion of Data Quality Assessment Classes**

Assessment Criterion	
Objective	Process
<ul style="list-style-type: none"><li>• Completeness</li><li>• Customer Support</li><li>• Documentation</li><li>• Objectivity</li><li>• Price</li><li>• Reliability</li><li>• Security</li><li>• Timeliness</li><li>• Verifiability</li></ul>	<ul style="list-style-type: none"><li>• Accuracy</li><li>• Amount of data</li><li>• Availability</li><li>• Consistent Representation</li><li>• Latency</li><li>• Response time</li></ul>

#### **2.4.4 Assessment Method**

According to Nauman and Rolker (2000) some factors to consider for conducting the assessment method of data or IQ measurement. These factors are score units and ranges, precision, practicality and technique. Factors considered are discussed below:

**Score Units and Ranges:** The system designer and the user must agree on a unit to measure, the criterion, and on a range to correctly score and usefully assess IQ criteria.

Table 2.5 shows the unit and the range of the assessment criterion classes.

**Table 2.5: Assessment Unit & Ranges of Assessment Classes**

Assessment Criterion Class	Unit	Range
Object-Criteria	Intuitive unit of expert input	1-10 or 0-1 , percentage
Process-Criteria	Seconds , Percentage	0 - $\infty$ , percentages between 0-100.

**Precision:** The precision of the IQ score is important for judging IQ. Imprecise assessment can either result in poor IQ or can lead to avoidance of high IQ. IQ scores could be imprecise for an imprecise definition of the assessment criteria. Therefore,



definitions of assessment criteria have to be precise. Further problems are distinct to the assessment criterion class. The respective requirements of precise assessment score for assessment criterion classes are given in Table 2.6.

**Table 2.6: Requirements of Precise Assessment Score for Assessment Criterion Classes**

Assessment Criterion Class	Requirements of Precise Assessment Score
Object-Criteria	<ul style="list-style-type: none"> <li>• Layout and format of information source <ul style="list-style-type: none"> <li>• Size of the sources</li> <li>• Sample size</li> <li>• Sample technique</li> </ul> </li> </ul>
Process-Criteria	<ul style="list-style-type: none"> <li>• Precision depends on the allocated time</li> <li>• The score is imprecise over time</li> </ul>

**Practicality:** Practical assessment results in precise assessment. An assessment method should be understood by the user and should be easily adapted to new sources and new requirements. Therefore, an assessment method should be as practical as possible. Requirements of practicality for the assessment criterion classes are given in Table 2.7.

**Table 2.7: Practicality Requirements of Data Quality Assessment for Assessment Criterion Classes**

Assessment Criterion Class	Requirements
Object-Criteria	<ul style="list-style-type: none"> <li>• Not to be costly</li> <li>• Not to be time consuming</li> </ul>
Process-Criteria	<ul style="list-style-type: none"> <li>• Not to be time consuming</li> </ul>

**Technique:** Assessment technique is one of the factors for the assessment method, like precision, practicality, range and unit. Techniques for assessment are not the same for each assessment class. Techniques vary for assessment criterion classes. Assessment techniques for assessment criterion classes are given in Table 2.8.



**Table 2.8: Techniques of Data Quality Assessment for Assessment Criterion Classes**

Assessment Criterion Class	Techniques
Object-Criteria	<ul style="list-style-type: none"><li>• Contract</li><li>• Parsing</li><li>• Sampling</li><li>• Expert input</li><li>• Continuous assessment</li></ul>
Process-Criteria	<ul style="list-style-type: none"><li>• Cleansing techniques</li><li>• Continuous assessment<ul style="list-style-type: none"><li>• Parsing</li></ul></li></ul>

According to Ge and Helfert (2008) criteria for evaluating the typical assessment methodologies are the definition of IQ dimensions, classification of IQ dimensions, model, tool and case study. The definition of IQ dimensions is identifying a definition of IQ dimensions from some perspective. Classification of IQ dimensions is used to compare the classification of dimensions in each methodology. The model demonstrates the theoretical basis of the methodology. The tool is used to validate the implementation of the methodologies.

#### **2.4.5 Assessment Matrices**

Matrices specify the characteristics of the system to be measured. Therefore, assessment matrices specify the characteristics to be considered for measurement of data or IQ. Assessment matrices are used for assessing objective or process criteria assessment.

According to Pierce (2004) control matrices are a concise way to link data problems with quality control that should detect and correct these data problems during the information manufacturing process. Elements of matrix can rate the effectiveness of the



quality check and reducing the level of data errors. Forms of the ratings are given in Table 2.9.

**Table 2.9: Types of Ratings Form for Data Quality Checking**

<b>Ratings Form</b>	<b>Function</b>
Yes/No	Address whether quality check is present or not
Category	Describes the level of error prevention, detection and correction. The level could be low, moderate or high
Number	Numerical assessment of the quality check of data for prevention, detection and correction
Formula	Measure the fluctuation of the reliability of a quality check and describe with a mathematical function

A number forms can be described with three functional forms. (Capiello et al., 2005; Papino et al., 2002). These functional forms produce numerical values for the assessment functions.

**Simple Ratio:** The simple ratio measures the ratio of required outcomes to actual outcomes. The range of the simple ratio is between 0 and 1. 1 represents the most desirable outcome and 0 represents the least desirable score. A simple ratio is suitable to obtain an aggregate measure along a single quality dimension.

**Min or Max:** Minimum or maximum operation can be applied for handling the dimensions that require the aggregation of multiple DQ indicators. It is also used to provide an aggregate value of DQ along a single dimension for a set of data. A significant example can be the timeliness dimension. Along a data set, the minimum value of timeliness is significant in order to understand the updating of the whole data set. Range for scoring min and max is between 0 and 1.



**Weighted Average:** In the multivariate case, an alternative to the min operator is a weighted average of dimensions. To obtain a normalized result, weight should range between 0 and 1 and their summation should evaluate to 1.

## **2.5 Data Quality Problem in IMS**

The extracted or inserted raw data may not be in the same data format or the source format could be different. Furthermore, the inserted or extracted raw data could be poor quality for incompleteness, inaccuracy, or inconsistent etc. Therefore, the inserted or extracted data from sources will be transformed and cleaned before loading into the data storage system. The quality of the inbound data of the organization may not have been checked before storing data in the data storage system of IMS. Ge and Helfert, (2008) assessed the data source point (database) of an organization where the measuring object was the stored data and the assessment unit was the percentage. DQ was assessed with objective DQ dimensions. The assessment was done by the software. They found DQ problems in the data of the data storage system of that organization.

Furthermore, a cooperative information system is a large scale information system that interconnects various system of different and autonomous organizations, geographically distributed and sharing common objectives. Data is the fundamental resources shared by the organizations. The efficacy of the cooperative macro-process depends on the capacity of the single autonomous system to share its data and the data of others. Vital factors of the cooperative information system are the definition of criteria metrics and methodologies to manage the quality of the data exported by each autonomous system, considering the selected metrics. There are often DQ problems in the data of the individual organization of the DSS of cooperative information system.



These DQ problems can be solved by periodical record matching and the query time improvement techniques (Milano et al., 2005).

**Periodical Record Matching:** A record matching algorithm is run on the inbound data sets stored by the organizations collected from multiple sources in the CIS. Hence, the analysed data sets of cooperating organizations are exchanged with each other within cooperative processes.

Record matching activity can be performed in two phases:

- A periodical record matching can be run in order to align different copies of the same entities that are present in data sources. The algorithm runs on data stored on distributed sources.
- The record matching activity also supports the query processing phase by identifying the same instances in query results returned by each data source.

**Query Time Improvement:** It will be done for the outbound data of cooperative information system. Outbound data is retrieved from the stored data by query processing. Data comparisons are performed at query processing time on data received as answers to queries. When organizations send a query request for the data, and all the sources of CIS may provide part of the answer. This means that different but semantically equivalent copies of the same data may be gathered. Copies are compared and only best quality results are returned. Providing best quality results as answers to queries avoids the spread of poor DQ throughout the system. Additionally, best quality result may be used to obtain an improvement of the overall quality of the system. Furthermore, data sources that have supplied poorer DQ can be notified with the best quality result constructed during query processing.



## 2.6 Data Quality Problem for Updating Data in IMS

Bruckner et al. (2001) works at the time related data integration for data warehouses. Continuous data integration is one of the most important requirements for time related data storage system. Chaudhuri and Dayal (1997) works for the data warehouse DSS. According to them, Data warehouse DSS store historical data, individual data record or consolidated data. Back end functions of data warehouses are data cleaning, loading and propagation. Cleaning function solves the errors and anomalies of data that comes from multiple sources. The load utilities load the cleansed data in the data warehouse. The loading process could be a full load. Completely erasing the contents of one or more tables and reloading with fresh data (full load). Another loading process is incremental loading. Only the updated tuples are inserted in the incremental load. The incremental load conflicts with on-going queries. So, it is treated as a sequence of shorter transactions. But, this sequence of transactions has to be co-ordinated to ensure consistency of derived data and indices with base data. After the completion of loading, propagation of data starts in the data warehouse.

Further, it needs the refreshment process for updating data in the data storage system. Refreshment process is done for the change data in the source. Change data comprise new source data (insertions) and modifications to existing source data (updates and deletions) (Mannino and Walter, 2006). According to Zdnek et al. (1998) missing and redundant data problem occurs in DSS at the time of update. Missing data problems can occur in two ways. Firstly, if data is not available in a particular data source at the moment of the data request; and secondly, when data in different data sources are available but at different moments, and in addition some of this data is needed sooner than it becomes available. Some of the data are available concurrently for certain



moments of data request in more data sources and its value are distinct. Therefore, the data redundant problem occurs.

Moreover, usage and evolution of quality oriented data warehouse DSS are described by (Vassiliadis et al., 2000). Refreshment process is one of the main data warehouse DSS processes for which quality is an important issue. Associated quality template includes the following quality dimension to the refresh process in data warehouse DSS.

**Data Freshness:** How old is the data (age of data).

**Data Completeness:** Ratio of the existing data in the data warehouse DSS and actual amount of data that should really be there.

**Data Coherence:** The respect of (explicit or implicit) integrity constraints from the data.

Quality dimensions associated with quality factors measure DQ in the data warehouse DSS. Quality dimensions are used to classify quality goals and factors into different categories. A quality factor is a special property or characteristic of the related object with respect to the quality dimension of the quality factor. It also represents the expected range of the quality values which may be any subset of the quality domain. Dependencies between quality factors are also stored in the meta data repository of data warehouse DSS where the stakeholders represent their quality goals. One can define the specific quality factors like: availability window, the extraction frequency of a source, estimated values for the response time of an algorithm, or the volume of data extracted each time. Quality factors can be distinguished between primary and derived quality factors along with design choices. Primary quality factors can be a simple estimation of



a stakeholder or a direct measurement. DQ dimensions of the primary quality factor are a subjective value directly assigned by the data warehouse DSS administrator. DQ dimensions of derived quality factors are computed as formula over other quality factors. Design choices are special kinds of quality factors, which aim to regulate the algorithm following the performance of each task in the data warehouse DSS.

Theodoratos and Bouzeghoub (1999) works on the currency quality factors for data warehouse DSS. The data currency quality goal is expressed by the currency constraint associated with every source relation in the definition of every input query. The upper bound in a currency constraint is set by the knowledge worker according to their needs.

## **2.7 Time Related Quality Aware Query**

Dong et al. (2006) works on quality aware queries. Some of the quality aware queries that are used are:

**Query1:** A currency bound query request. The query request is used to address the quality of the data of a query result. Example of a currency bound query request is,

```
SELECT NAME
FROM BOOK B, REVIEWS R
WHERE B.ISBN = R.ISBN
CURRENCY BOUND 10 MIN (B, R)
```

In this query, the currency bound quality clause indicates that the input tables B and R cannot be more than 10 minutes out of date data.

**Query 2:**“Select all information about medicines for „headache“, in which amount information was last updated after „2006-04-01 12:00:00+00:00“, from relation Medicine.”



```

SELECT *
FROM Medicine
WHERE Category = 'headache'
WITH QUALITY AS
Last Update Time (Amount) > '2006-04-01 12:00:00+00:00'

```

In this query the „Amount“ information of a medicine that has been last updated before or on „2006-04-01 12:00:00+00:00“ are not acceptable.

**Query 3:**“Select the medicine’s name, price and the timeliness score of price from relation Medicine, for medicine „headache“, with the timeliness score for „amount“ information greater than 0.60.”

```

SELECT Name, Price, TIMELINESS (Price)
FROM Medicine
WHERE Category = 'headache'
WITH QUALITY AS
TIMELINESS (Amount) > "0.60"

```

Timeliness of „Amount“ attribute greater than .60 is the quality information for this query.

Quality aware query is used to ensure the quality of information. Timeliness of data quality can vary in heterogeneous IMS for the variation of the refreshment processing time and the waiting time of data in IMS. Therefore, this quality aware query can use for measuring the quality of information of heterogeneous IMS. For example: let, there are IMS1 and IMS2. Refreshment processing time and waiting time of IMS1 is less than the refreshment processing time and waiting time of IMS2. It is shown in chapter 4 that timeliness depends on the refreshment processing and waiting time. Now, if timeliness of data of IMS1 and IMS2 is .5 and .4 respectively and the quality aware



query is set that  $\text{timeliness} > .4$  is good quality information, then, information of IMS1 and IMS2 will be good and bad quality information respectively.

## **2.8 Trade-off between Timeliness and other Objective Data Quality Dimensions**

There is a deadline of transactions in the field of information system. The deadline of a transaction indicates that the transaction needs to be completed before a certain time in future. Ulusoy (1995) discusses different types of transaction deadlines. Transaction deadlines:

Hard deadline transactions are associated with strict deadlines and the correctness of transaction operations depends on the time at which the results are produced. The system must provide schedules that guarantee deadlines. Nuclear power plants, air traffic control systems, and process control systems are some example of applications that process hard deadline transactions.

Soft deadline transactions are scheduled based on their deadlines, and satisfaction of deadlines is still the primary performance goal in scheduling transaction. In this case, there is no guarantee that all deadlines will be met. A soft deadline transaction is executed until completion regardless of whether its deadline has expired or not. The banking system and airline reservation systems usually process soft deadline transactions. When a customer submits a transaction, if the system cannot generate a response to the transaction within its deadline the customer prefers to get a late response to not getting one at all.



Firm deadline transactions do not carry strict deadlines. This means that missing a deadline may not result in a catastrophe, but unlike soft deadline transactions, they are aborted by the system once their deadlines expire. Typically, no value will be imparted to the system if a firm deadline transaction misses its deadline. Stock market trading is one example of applications supporting firm deadline transactions. If, for instance, a transaction is submitted to acquire the current price of a particular stock, the system should either return the result in a specified time period or not perform the operation at all.

Hu et al. (2007) addresses the necessity of a time constraint query request to get the information in time. Long running queries and unpredictable query response time use the following approaches:

**Optimize for the first few rows:** Users can indicate that they want to optimize the query to return the first few rows as soon as possible.

**Optimize for the top- K rows:** In this approach, the user is interested in only the top-K candidates of the sorted result.

**Compute the approximate result:** This approach speeds up query processing by working only on partitions of data and hence returns an approximate result.

The end user has time constraints and needs the query results within certain time bounds. Hu et al. (2007) introduced the concept of both soft time constraints, where the end user only gives directives on the desired time for the query to return results, and hard time constraints, where the query is guaranteed to return results within the



specified time. For a time-constrained query, the query results can be approximate or partial. The query can sample a portion of the data that is queried and return approximate results based on the sample. Alternately, it can choose to return only a partial subset of the results (first few rows or top K rows in case of sorted result) with the guarantee that they are accurate. An example of the time constraint query requests is:

```
SELECT AVG (salary)
FROM employees
SOFT TIME CONSTRAINT (50)
WITH APPROXIMATE RESULT;
```

The user indicates that the specified query should be completed with approximate results in 50 seconds. The similar query request is used for the hard time constraints.

Vrbsky (1996) mentioned a model for approximate query processing in real time DSS. A temporal data model for approximate query processing has been presented. Time constraint queries are used to get the approximate query result. Vrbsky and Tomic (1998) examined the performance of approximate query processing (AQP) to determine its effect on satisfying the timing constraints of real time DSS. Hsien et al. (2001) developed a new method to measure the quality of each data set of a query in a fuzzy relational database. A model is developed for the quality of query answers. The quality of an answer is viewed as how much information is provided and how much extra information is needed so that it will be a complete answer to the query.

Capiello and Helfert (2008) analysed the presence of trade-offs between availability and timeliness in the data redundant system. The time interval between the realignments and late propagation of data may cause DQ problems. It is stated that DQ



increases with synchronization frequency. It makes a positive effect on DQ by the immediate propagation of data changes through all the sources in the system. In the data redundant system, timeliness is considered as a critical dimension. Timeliness depends on two factors: the time instant in which data are inserted in the sources or transferred to another system and data volatility. If the frequency of synchronization is very high, the system is able to guarantee updated information with a high degree of timeliness while the availability of the system is low, since sources are often locked for loading data. On the contrary, if the frequency of synchronization is low, the availability of data in the system is high.

DQ dimensions are correlated with each other. If one dimension is considered more important than the others for a specific application, then the choice of favouring it may imply negative consequences for the others. Batini and Scannapieco (2006) describes the environment for the trade-off between the dimensions and the occurrence of trade-off between DQ dimensions. Basically, the trade-off is done between the time related dimension (timeliness) and the non-time related dimensions like completeness, accuracy and consistency. Environments for the trade-off between timeliness and other DQ dimensions are web application, where time constraints are very stringent and the DQ checking activities in an application needs time and timeliness is negatively affected.

An internet query system is used to query the World Wide Web by finding data sources relevant to a given query and retrieving data. Sampaio et al. (2005) addresses DQ issues such as timeliness and accuracy of data resulting from internet query processing.



According to Pernici and Scannapieco (2003) a web based information system uses internet web technologies to deliver information and services to users and other information systems. The main purpose of web information is to publish and maintain data by using hypertext based principles.

Some reasons for DQ problems in web information systems are given below:

- Information systems on the web need to publish information in the shortest possible time after it is available from information sources.
- Lack of accurate design of data structures of good navigational paths between pages.
- Costly and lengthy verification of the data to be certified.

Trade-offs between objective DQ dimensions and timeliness of information also occur in web information systems.

## **2.9 Summary**

The purpose of this literature review is to discuss the existing research relevant to this study. The IMS is the principal area of this research. Therefore, relevant research such as the structural and the operational elements of the IMS are discussed. There are DQ problems in the IMS. Hence, the most usable DQ dimensions are identified and definitions of these dimensions discussed. Data of the IMS are assessed for the measurement of DQ. Consequently, a review was conducted for different assessment dimensions. DQ problems in the IMS are reviewed as relevant to this research. Finally, it is shown that there is a trade-off between timeliness and objective DQ dimensions in IMS. It means that if timeliness is good, DQ for the objective DQ dimensions are not good. Therefore, DQ changes with timeliness. It is learnt from the literature review that



DQ changes with timeliness. However, there is no research that shows that how DQ changes with timeliness. Therefore, it is shown in this research. Further, improvement of DQ with timeliness is also shown in this research by comparing the DQ of heterogeneous IMSs.



### 3 Heterogeneous Information Manufacturing System (IMS) and Data Quality Constraints for IMS

---

#### 3.1 Introduction

Heterogeneous entity is composed of dissimilar parts. Hence, the constituents of an entity will be of different kinds. Heterogeneity considers all types of semantic and technological diversities among the systems used in modelling and physically representing data such as hardware, software, material, database management systems, programming languages, and operating system etc. (Wang et al., 2001).

Some of the factors adopted for classifying the heterogeneous IMS are shown in Figure 3.1.

		Elements	Factors
Factors for Heterogeneous Information Manufacturing System	Structural Elements	Integration	Degree of Integration
		Data Storage System	Execution Method of Refreshment & Query Function
			Execution Method of Tasks of Refreshment Function
			Execution of Number of Tasks for Refreshment Function
	Operational Elements	Machine	Capacity
		Material	Volume of Data
			Change Frequency of Data
		Refreshment Processing	Refreshment Frequency Method
		Query Processing	Query Method

Figure 3.1: Factors for Classifying Heterogeneous Information Manufacturing System



The adopted factors for heterogeneity of IMS are divided into factors for the structural and operational elements. The factors of the data storage system and integration of source elements make the IMS heterogeneous. The factors of the integration and data storage system elements are degree of integration, the execution method of the refreshment and query function, execution method of the tasks of the refreshment function and the number of tasks for refreshment function respectively. The existing data storage system (single DSS, cluster DSS and 2-DSS) and the newly modelled data storage system (3-DSS) by (Islam and Young, 2013) are used for showing the data quality variations for the factors of data storage system. These existing and newly modelled data storage system are discussed in this chapter. Similarly, heterogeneity of IMS varies for the factors of the machine, material, refreshment processing and query processing operational elements. Capacity, volume of data and change frequency of data, refreshment frequency method and query method are the factors for the machine, material, refreshment processing and query processing respectively.

According to (Capiello et al., 2005; Capiello and Helfert, 2008), DQ varies in IMS for the continuity of available data, uniformity of available data and currency of available data. Therefore, these constraints will be further discussed in this chapter.

## **3.2 Heterogeneity of the Structural Elements of Information**

### **Manufacturing System**

Factors of the integration and data storage system elements have different dimensions. IMS is heterogeneous for these dimensions. The data storage system varies for the different dimensions of the factors of the data storage system. Therefore, different dimensions of the factors of elements of the IMS will be discussed.



### 3.2.1 Degree of Integration of Sources

The degree of integration of the sources is incremental in the IMS. Therefore, the degree of integration of the sources for storing data in the DSS could be lowest to highest degree integrated (Capiello et al., 2005). The degree of integration for various sources or channels in the IMS can be described as:

**Lowest Degree of Integration:** Different or uniform software applications of multiple channels or sources store the different functionalities data in the separate DSS for the lowest degree of integration. As, the degree of integration of sources or channels is low category, the updated data of the separate DSS of the IMS are to periodically align with a given refresh period. Therefore, the manipulated data in each individual DSS of the IMS cannot be updated at the same point of time within the same refresh period.

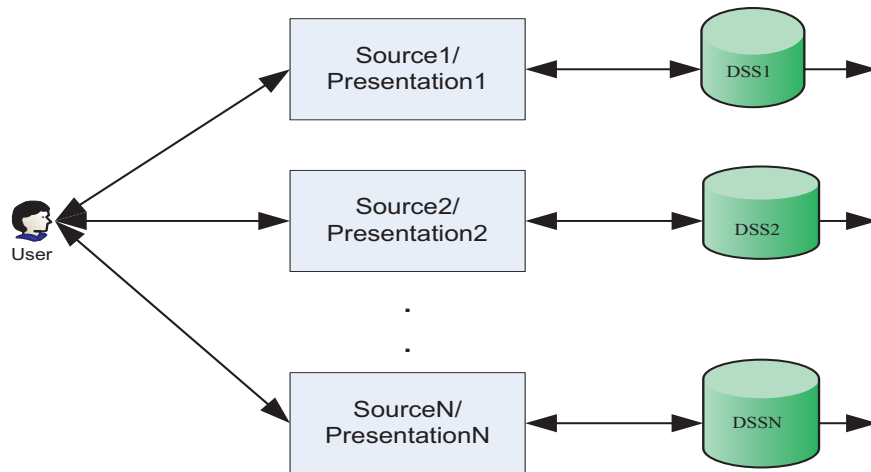
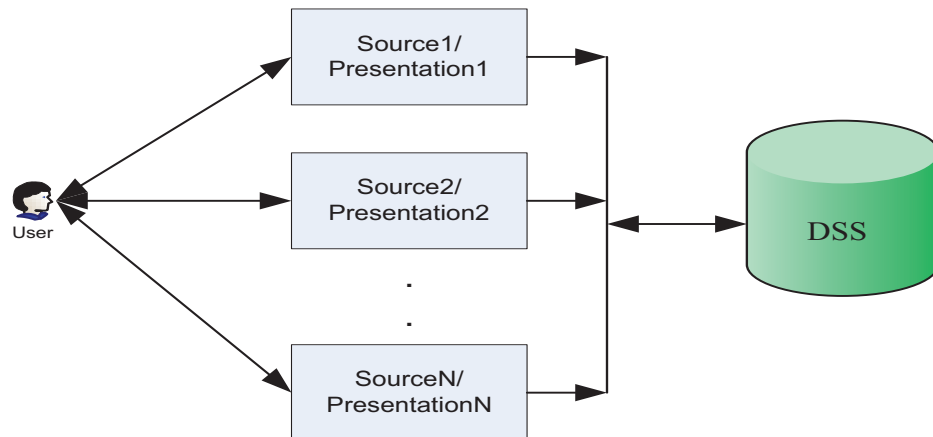


Figure 3.2: Lowest Degree Integrated Information Manufacturing System

**Highest Degree of Integration:** Each source or channel of the system is integrated with the highest degree to store data in the DSS of the IMS. Therefore, each of the uniform or non-uniform software applications can store data and provide information in



one single or virtual single DSS. Refreshment process works in only the single or virtual single DSS for updating the manipulated data in the IMS. As a result, each user accessing any functionality from any channel or any sources will read and update the same information.



**Figure 3.3: Highest Degree Integrated Information Manufacturing System**

### 3.2.2 Types of DSS

In the IMS, method is the way or technique of processing and delivering information (Wang et al., 2008). Manipulated data must be available in the IMS of the organizations to provide information support. The refreshment and query functionalities of DSS are applied for making data available in the IMS and delivering information from IMS. Types of data storage system differ in the IMS for the execution method of both functionalities of the DSS and the tasks of the refreshment function. The execution methods of the DSS functionalities and the tasks of the refreshment function are:

**Sequential Method:** The tasks of the refreshment function and the query function execute sequentially as serial manner.

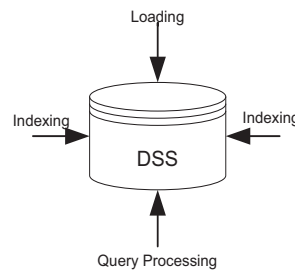


**Simultaneous Method:** Both the tasks of the refreshment function and the query function execute simultaneously as parallel manner.

The execution method of the tasks of the refreshment and query function and the execution method of the tasks of the refreshment function could be (sequential, sequential), (sequential, simultaneous), (simultaneous, sequential) and (simultaneous, Simultaneous) in the different types of DSS. Furthermore, the number of tasks for the refreshment function could be varied in the DSS. Various types of data storage systems used in the IMS are now looked at in more detail.

### 3.2.2.1 Single DSS

The single DSS is discussed in detail in (Capiello et al., 2005). Data comes from multiple sources are stored in the single DSS. Source data may come from multiple functional areas or from multiple channels. The channels and functional areas are not the same for all organizations. For example, the functional area for a bank is trading,



insurance, credit card etc. and the channel could be internet, ATM booth, or branch. Loading and indexing tasks execute in this DSS for making data available in IMS. The tasks of the refreshment and query function of the data storage system run in the single DSS. Therefore, execution method of the tasks of the refreshment and query function is sequential. As a result, outbound data services (information support, data analysis etc.) in the IMS may be non-continuous for the sequential execution method of the tasks of the functionalities. The degree of integration process of the single DSS is highest.

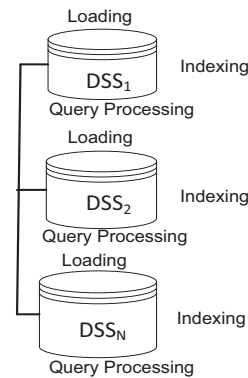


Additionally, it does not need the propagation of manipulating inbound data for the highest degree integration of source data to be stored in the single DSS.

### 3.2.2.2 Cluster DSS

According to Capiello et al. (2005) and Pape and Gancarski (2007) data from the sources (same functional area but different

channel and vice versa) are stored in the DSS as the cluster. There is no master-slave DSS in the cluster DSS (Capiello et al., 2005). Therefore, the degree of integration process is lowest. Pape and Gancarski (2007) has the master-slave DSS in the

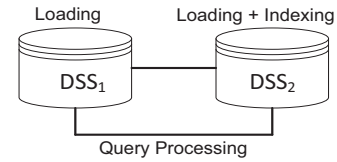


cluster. For this reason, the degree of integration process could be the highest. Each individual DSS of both types of cluster DSS conduct the DSS functionalities. If data is manipulated in any of the DSS of the cluster, loading and indexing mechanism process the manipulated data and propagates the manipulated data to the other DSS of the cluster. The execution method of the DSS functionalities in the cluster DSS is simultaneous. This means, when one DSS of the cluster executes the refreshment function another DSS can execute the query function at the refreshment period. On other side, the tasks of the refreshment function execute sequentially in each individual DSS of the cluster for the availability of data in IMS.



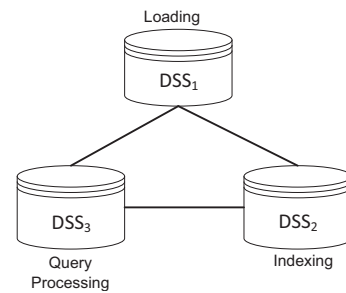
### 3.2.2.3 2-DSS

Santos et al. (2008) works on 2-DSS for providing information support as quickly as possible. In 2-DSS, data that come from multiple sources are temporarily refreshed (only load but no indexing) in one DSS. Query function is done sequentially with the refreshment (only load but no indexing) function on that DSS. After a certain time, data of the temporary DSS is loaded and indexed to the permanent DSS for the deterioration of query response time. Permanent DSS of the 2-DSS work like the single DSS. The tasks of the refreshment function and the query function of data storage system work sequentially just like the single DSS. The degree of integration process is considered higher for the 2-DSS.



### 3.2.2.4 3-DSS

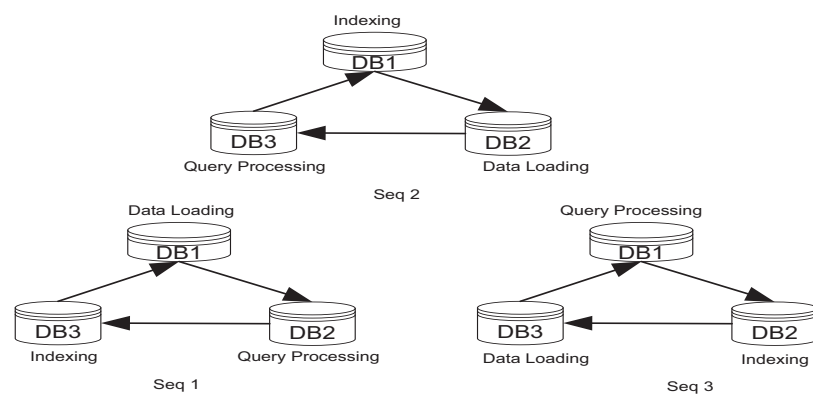
3-DSS is described in (Islam and Young, 2013). In 3-DSS, three individual DSS are mutually interconnected with each other. It works similarly to a virtual single DSS. Loading and indexing tasks execute in the 3-DSS for the refreshment function. The tasks of the refreshment and query function of the data storage system work in three individual DSS simultaneously. Data loading and indexing with updated data propagation tasks work in two individual DSS of 3-DSS at the same time. Another DSS of 3-DSS executes the task of the query function. After each successive period, the tasks of the refreshment and query function of the data storage system will interchange with cyclic order. As, propagation of manipulated data on the DSS is carried out simultaneously at the working period of the task of the





functionalities, there may not be propagation delays. Therefore, the 3-DSS will hold exactly the same data. Furthermore, as the 3-DSS work like a virtual single DSS, the degree of integration process of sources is higher for the 3-DSS.

Now, execution process of 3-DSS will be described. Suppose, DB1, DB2 and DB3 is three individual DSS for the 3-DSS. These three DSS are mutually interconnected with each other. Now, if DB1 store some data from operational data sources, DB2 and DB3 must have the same data. The tasks of the refreshment and query function work simultaneously in these three data storage systems. There must be a synchronization of starting and finishing time of the tasks of the refreshment and query function of these three data storage systems. Manipulated data from the operational data sources will load into one database. At the same time, another database will index and update data propagation task for synchronizing data with other two DSS and the third will be used for query processing within the certain time. The indexing and the query processing database lead the process. When the indexing with propagation of the updated data and the query processing are finished, the tasks of refreshment and query function of the data storage system will interchange with cyclic order. The rotation sequence for the interchanging process is shown in Figure 3.4.



**Figure 3.4: Rotation of the Tasks of Functionalities in 3-Data Storage System**



In this paragraph, regulating procedure of the tasks of the DSS functionalities in 3-DSS will be discussed. 3-DSS executes three individual DSS simultaneously in the IMS. Therefore, it needs the regulator to control the three DSSs that are tied together. As the tasks of the functionalities of the data storage system work in three individual DSS of the 3-DSS, there must be coordination and communication among the tasks of the functionalities. Coordination of the tasks of the functionalities of the DSS can operate the 3-DSS based IMS perfectly. Therefore, it needs a third party agent that could be called the Synchronizing Agent of the 3-DSS for coordinating the tasks of the functionalities of the 3-DSS. Coordination of the tasks of the functionalities (loading, indexing, query processing) in the 3-DSS depends on communication among tasks. This coordination of tasks of the functionalities helps the interchanging of the tasks of functionalities among DSSs of the 3-DSS. Interchanging of tasks of functionalities of the 3-DSS relies on completion of the indexing and the query processing tasks. Therefore, the indexing and the query processing tasks act as a controller for the interchanging process of the tasks of the functionalities of the 3-DSS. The algorithm for regulating the tasks of the functionalities of 3-DSS is given in Figure 3.5.



***Regulator (DSS<sub>1</sub>, DSS<sub>2</sub>, DSS<sub>3</sub>)***

*(Loading), (Indexing and Propagation) and (Query Processing) is represented as L, Ix and IS respectively*

*3-DSS is controlled by a controller thread represented as CT*

*Step 1: Create Thread 1, Thread 2, Thread 3 and Controller Thread as T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub> and CT*

*Step 2: T<sub>1</sub> → Ix (), T<sub>2</sub> → IS (), T<sub>3</sub> → L ()*

*Step 3: T<sub>1</sub> → L (), T<sub>2</sub> → Ix (), T<sub>3</sub> → IS ()*

*Step 4: T<sub>1</sub> → IS (), T<sub>2</sub> → L (), T<sub>3</sub> → Ix ()*

*Step 5: CT → Execute Synchronizing Agent Algorithm*

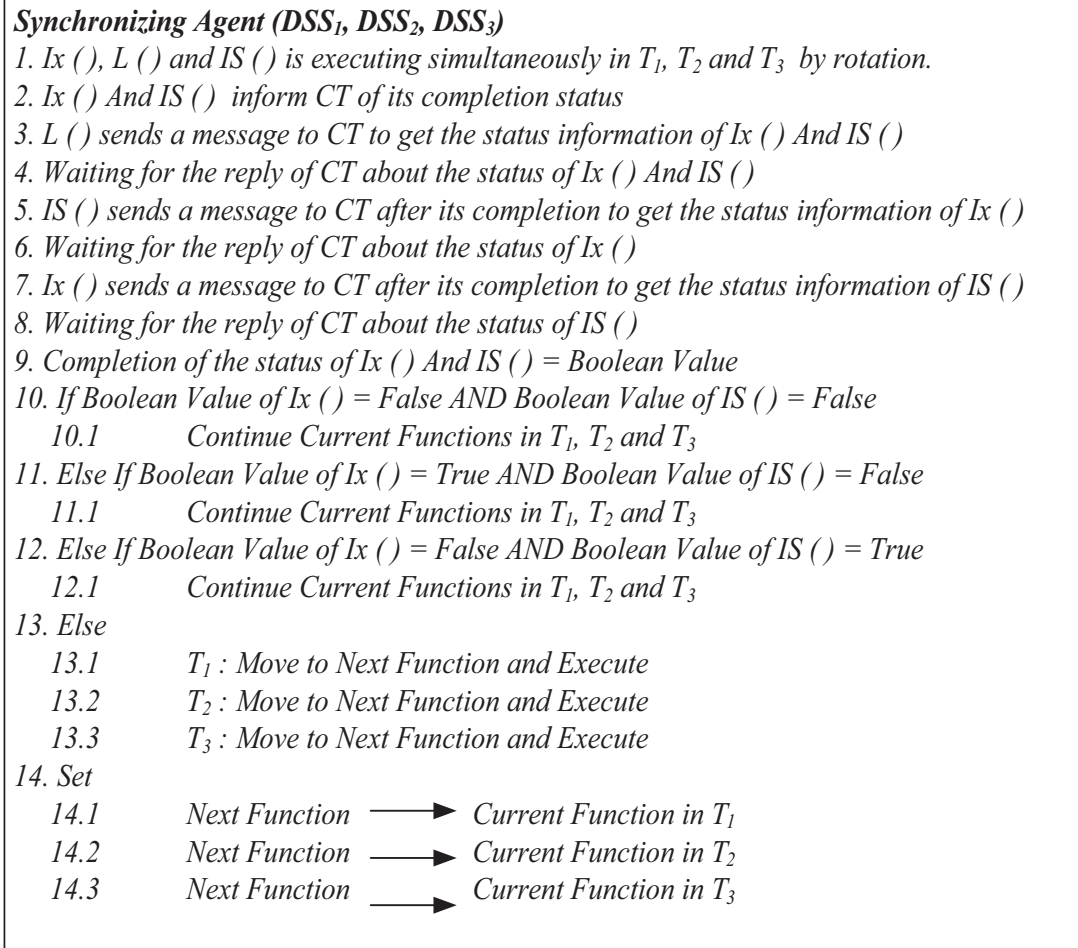
*Step 6: Go on step 2 and repeat from step 2*

**Figure 3.5: Regulator Algorithm for 3-DSS**

In the regulator algorithm of the 3-DSS, four threads are created for the execution of the operation of 3-DSS. Thread 1, thread 2 and thread 3 are created for the simultaneous operation of the tasks of the functionalities of the 3-DSS in the three individual DSS. The controller thread CT is constructed for the execution of the Synchronizing Agent. The Synchronizing Agent algorithm is shown in Figure 3.6. Thread 1, thread 2 and thread 3 works simultaneously, so, when thread 1 executes the indexing function, thread 2 and thread 3 will execute the query and loading function respectively. This will continue until thread 1, thread 2 and thread 3 get a message from the Synchronizing Agent of the controller thread to change their tasks of the functionalities. If thread 1, thread 2 and thread 3 get a message from the Synchronizing Agent of the controller thread to change their activities, then, thread 1 executes the function for the loading task, thread 2 and thread 3 executes the functions for indexing and query processing tasks respectively. These activities of threading will continue until the threads do not receive any more messages to interchange their activities. As soon as, each thread gets the message to interchange their activities, thread 1 will start query processing, thread 2 will start loading of data and thread 3 will start the indexing



task. The sequence of activities among the threads will continue until the system is stopped by the user. Details of the Synchronizing Agent algorithm are given in Figure 3.6.



**Figure 3.6: Synchronizing Agent Algorithm for 3-DSS**

Figure 3.6 presents the algorithm showing the interchanging process of the tasks of the functionalities of the 3-DSS. It shows, how the tasks of the functionalities of the 3-DSS communicate with each other for providing their service rotationally in three individual DSS of the 3-DSS. According to the regulator algorithm of the 3-DSS, the tasks of the functionalities of the 3-DSS (indexing, loading and query processing) execute simultaneously in three threads by rotation. Therefore, each individual DSS of the 3-




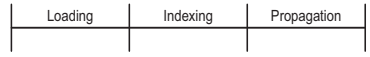
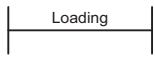

DSS change after a certain period of time. Query processing and indexing tasks do not stop before the completion of tasks. Therefore, the indexing and the query processing tasks can be referred to as dependent tasks. However, it is possible to stop the loading of data at any time. Therefore, this task could be referred to as the independent task. The interchanging process of the tasks of the functionalities in the 3-DSS depends on both the indexing and the query processing tasks. Line 1 of algorithm indicates that step 1, step 2 and step 3 of the regulator algorithm will be executed simultaneously by rotation. In line 2, the function of the indexing and the query processing tasks will inform their current status to the controller thread. Then, the function of the loading task will send the message to the controller thread to know the current status of the indexing and the query processing function in line 3. The controller thread will deliver a reply about the status of the indexing and the query processing in line 4. In line 5 and 6, the query processing function sends a message to the controller thread to know the status of the indexing function after the completion its task and wait for the reply. Similarly, in line 7 and 8, the indexing function will do the same and wait for the reply about the status of the query function. Line 10 to line 13 shows whether the tasks of the functionalities of the 3-DSS will interchange or not. If the completion status of the query processing or the indexing function is false, the tasks of the functionalities of the 3-DSS will not be interchanged. So, from line 10 to line 12, the current function is continued in thread 1, thread 2 and thread 3. In line 13, the completion status of both the query processing and the indexing function is true, so, the tasks of the functionalities of the 3-DSS are interchanged. For this reason, the current function of each thread is stopped and moved to the next function. The current function move to the next function means that in the regulator algorithm, indexing, loading and query



processing function execute in step 1, step 2 and step 3 respectively in thread 1; query processing, indexing and loading function execute in step 1, step 2 and step 3 respectively in thread 2 and loading, query processing and indexing function execute in step 1, step 2 and step 3 respectively in thread 3. Now, if the current function in step 1 of thread 1, thread 2 and thread 3 are indexing, query processing and loading function respectively, the next function will be the function of thread 1, thread 2 and thread 3 of step 2 and so on for step 2 and step 3. Therefore, the next function will be prepared for execution, and it will be executed as the current function. Finally, in line 14, the next function is set and executed as the current function in thread 1, thread 2 and thread 3. The whole process executes repeatedly to continue the interchanging of the tasks of the functionalities in three individual DSS simultaneously in the 3-DSS.

The dimensions that are found for the factors of the heterogeneous DSS component are given in Table 3.1.

**Table 3.1: Dimensions for the Factors of the Heterogeneous DSS**

DSS \ Factor of DSS	Method		
	Execution Method of Refreshment & Query Function	Execution Method of the Task of Refreshment Function	Execution of the Number of Tasks for Refreshment
<b>Single DSS</b>	Sequential	Sequential	
<b>Cluster DSS</b>	Simultaneous	Sequential	
<b>2-DSS</b>	Sequential	-----	
<b>3-DSS</b>	Simultaneous	Simultaneous	



### **3.3 Heterogeneity of the Operational Elements of IMS**

The factors of the operational elements of IMS such as machine, material, refreshment processing and query processing have different dimensions. IMS is also heterogeneous for these dimensions. Different dimensions of the factors of the operational elements of the IMS are now discussed.

#### **3.3.1 Machine**

The data processing capacity of the machine depends on the transaction throughput of the machine. The transaction throughput of a machine can be expressed as a number of transactions per second (Rizvi and Chung, 2010). If the processing capacity of the hardware is lower than the software, then, the transaction throughput of the machine will be the data processing capacity of the hardware. Furthermore, if the processing capacity of the software is lower than the hardware, then, the transaction throughput of the machine will be the data processing capacity of the software. For example: Database software such as ORACLE and SQL SERVER has a powerful data processing function (Zhou and Ding, 2006). However, data processing capacity of a hard disk could be lower than the data processing capacity of the database software. Therefore, the transaction throughput of a machine will be the data processing capacity of the hard disk (Rizvi and Chung, 2010; Santos et al., 2006; Zhou and Ding, 2006). The data processing capacity of the machine varies with the variation of the hardware and software in the IMS. If the hardware is fixed and the processing capacity of the hardware is higher than the software, then, the transaction throughput of the machine will be varied only for the variation of the software. Hence, high and low capacity machine will be determined by the variation of the software. Moreover, if the software is fixed and the processing capacity of the software is higher than the hardware, then,



the transaction throughput of the machine will be varied only for the variation of the hardware. High and low capacity machine will be determined by the variation of the hardware.

### **3.3.2 Material**

The material is the raw data of the IMS in an organization. Raw data in the IMS of an organization will be both new raw data and existing raw data. Flow of data from source to the data storage system means that data is not yet stored in the DSS of the IMS. This data can be introduced as the new raw data. On the other hand, if data is already stored in the DSS of the IMS as information it can be recognized as existing raw data.

#### **3.3.2.1 Types of Storage Data**

The IMS can store different types of data as raw data for manufacturing information. The IMS is versatile such as the web based IMS, the co-operative IMS or the IMS for shops, supermarkets, hospitals, and telecommunication etc. The IMS of these organizations needs to work with digital data. Any particular type of data may not be suitable for all types of IMS. As a result, considering the versatility of the IMS, data can be divided into three types (Batini and Scannapieco, 2006).

**Unstructured Data:** When data are expressed in natural language and no specific structure and domain types are defined it is called unstructured data. Human voice is one example of unstructured data.

**Semi-Structured Data:** Data of a semi-structured format is not fixed. It has some degree of flexibility. XML is the mark-up language commonly used to represent semi-structured data.



**Structured Data:** When each data element has an associated fixed structure is called structured data. Data of relational tables are an example of the structure data. Structured data is more manageable than other types of data. Therefore, both semi-structured data (XML document) and unstructured data (voice) can be stored as structured format in the relational DSS of the IMS.

According to Mannino and Walter (2006) scope criteria refer to the volume of data from sources involved in a refresh job. Therefore, the scope indicates the volume of data refreshed in an individual refreshment period. The volume of data in the period of refreshment could be high or low volume of data. The data volume of high and low is relative. For example: Suppose, existing data of an IMS is fixed. The IMS has  $(S + 1)$  sources and each source have the same volume of data. So, data can be refreshed from  $S$ ,  $(S - 1)$  and  $(S + 1)$  number of sources. Now, if the data is refreshed from  $(S - 1)$  numbers of sources, the refreshed volume of data from  $S$  number of sources will be the high volume of data. However, if the data is refreshed from  $(S + 1)$  numbers of sources, the refreshed volume of data from  $S$  number of sources will be low volume of data.

#### **3.3.2.2 Change Frequency of Storage Data**

Structured data of the IMS can be categorized by the change frequency of data. This change frequency of data depends on the expiration time of the data. Some data may expire after an hour, a day, a week, a month or even a year and some data may expire within a second or a minute. Therefore, considering these limits of the expiration time, the data are to update for the quality purpose of the data in the IMS. According to Batini and Scannapieco (2006) structured data of the IMS can be categorized as:



**Long-Term-Changing Data:** This is data that has very low change frequency. Examples are addresses, currencies and hotel price lists. The concept of low frequency is domain dependent; in an e-trade application, if the value of a stock quote is tracked once an hour, it is considered to be a low frequency change, while a shop that changes its goods weekly has a high frequency change for clients. Therefore, this type of data can also be called non-frequently changing data.

**Frequently-Changing Data:** Data that has an intensive change is frequently changing data. Real time traffic information, temperature sensor measures, and sales quantities could be examples of frequently changing data. The changes can occur with defined frequency or they may be random.

Change frequency of newly inserted data will fall into one of the above categories or mixed categories. Similarly, modification (delete, update) of existing data will be carried out considering the change frequency of the data. Frequent modification tasks must be executed for frequently changing data. Modification of long term changing data need not be changed frequently. According to Ballou et al. (1998) volatility of data relies on the change frequency of data. Measurements of the timeliness of data units depend on the change frequency of data (non-frequently changing/frequently changing).

### **3.3.3 Refreshment Processing**

The refreshment process of an organization can be either source driven or data storage system driven. The source driven refreshment process is triggered by changes made in



the sources. Data storage system driven refreshment process is triggered from the data storage system after changes have been made in the sources. These processes are characterised in the organization by using the refreshment frequency method. The refreshment frequency method refers to the number of times per time interval that each refresh job is performed. These refresh jobs are done with either incremental batch refresh mechanism or incremental continuous feed refresh mechanism. These mechanisms only aggregate the new insertions and the updates from the sources or the organizational database. Further, the source driven system executes the refreshment process frequently. The frequent refreshment can be referred to as the continuous refreshment frequency method. The DSS driven system executes the refreshment process periodically. The periodic refreshment process executes after a certain time interval. Therefore, periodic refreshment process is referred to as the non-continuous refreshment frequency method. Mannino and Walter, (2006) shows some organization of continuous and non-continuous refreshment frequency method. Some of the organizations applied continuous refreshment frequency method and some applied non-continuous refreshment frequency method for executing the refreshment process. To these organizations, non-continuous refreshment is done either at 5 to 10 minute interval, daily intervals, weekly interval or monthly interval.

#### **3.3.4 Query Processing**

The query retrieves the data from the DSS of the IMS as the outbound data for information support. Information needs to be delivered in real time, right time, and useful time or without considering the time. Therefore, the query method is important for query processing. The query method could be non-time constraint or time constraint query method. The time constraint query request is involved with delivering the



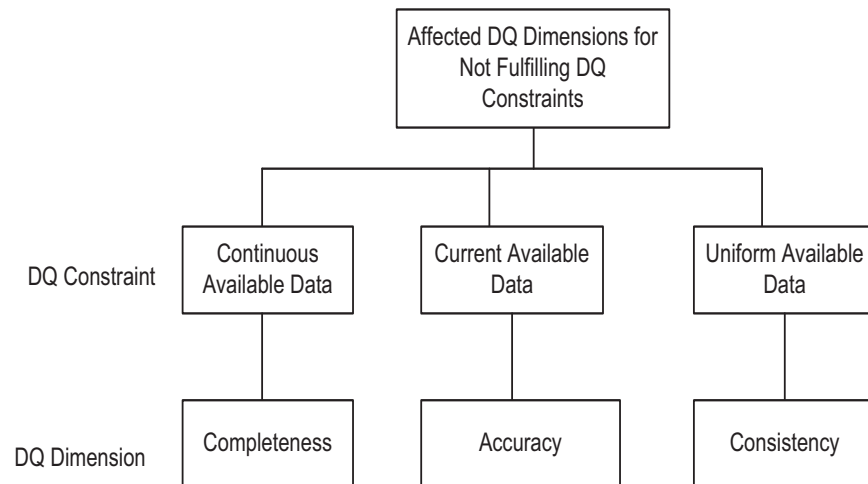
information just in time. The time constraint query method has to get the data within time. Hence, a time clause is used in the time constraint query request (Hu et al., 2007). The time constraint query request is further discussed in details in chapter 2. Non-time constraint query method is used for data delivery from the IMS when data do not need to be delivered at the right time, useful time or real time. Therefore, it does not need to use the time clause as in non-time constraint query request.

### **3.4 Data Quality Constraints in IMS**

DQ of the IMS varies for the continuity of available data, currency of available data and uniformity of available DQ factors (Capiello et al., 2005; Capiello and Helfert, 2008). Continuity of available data does not mean the continuous required available data in the IMS. Rather, it means both continuous and non-continuous availability of required data for manufacturing information. Similarly, currency of available data and the uniformity of the available data have two states. Whether the currency of available data will be current or non-current. Uniformity of available data ensures the uniform and non-uniform available data or data set in the IMS. According to Capiello et al. (2005) and Capiello and Helfert (2008) uniform information, continuous available data and current available data are the constraints of good IQ in IMS.

One reason for poor DQ dimensions is not fulfilling the DQ constraints of the IMS. These DQ constraints effect on objective DQ dimensions for timeliness DQ dimension in the IMS. The affected DQ dimensions for the DQ constraints are shown in Figure 3.7.





**Figure 3.7: Affected Data Quality Dimensions for not Fulfilling Data Quality Constraints**

**Continuous Available Data:** Availability of data could be continuous or non-continuous. Continuous available data means whether the user gets required available data at any moment of time or not. If the user gets required available data at any moment of time, availability of data is continuous. It is vice versa if the user misses required available data at any moment of time. Causes of non-continuous availability of data in the IMS are locking of data at the time of synchronization or refreshment process, not complete the execution of refreshment process and the unavailability of DSS of the IMS (Capiello and Helfert, 2008). Therefore, non-continuous available data just produce approximate or incomplete data. As a result, continuous available data can ensure the completeness of data for producing information.

**Current Available Data:** Continuous available data in the IMS is not enough for good IQ. Additionally, available data have to be up to date or current. Otherwise, the IMS can produce obsolete information. Current available data does not mean that data is available or not in the DSS of the IMS. Rather, it means whether data are current or not. Currency of data is measured by the timeliness value. If, the data timeliness value is smaller than the required timeliness value, data will be inaccurate. On the other hand,



if, the data timeliness value is greater than the required timeliness value, data will be accurate. Therefore, current available data can ensure the accuracy of the data in the IMS.

**Uniform Available Data:** This constraint ensures that the IMS will deliver exactly the same information for each query request sent at the same point of time. For example: if  $Q_1, Q_2, \dots, Q_n$  send the query request to the IMS at the time point  $t_1$ , each individual query request has to deliver exactly the same information. But, if each individual query request gets different information from the IMS this means that the system could not provide uniform information support. Reason for non-uniform information support is that data may be propagated from the master DSS to the slave DSS in the IMS. Therefore, data could be available all the time in the DSS of the IMS. However, when newly loaded or de-loaded data are to be propagated from the master DSS to the slave DSS, the IMS may not provide the same set of data for the same query request at the same time. As a result, data or the data sets can conflict with each other. Therefore, it may cause consistency DQ problem.

### 3.5 Summary

In this chapter, the heterogeneous IMS and the constraints of DQ for heterogeneous IMS were discussed. Different parameters of the factors of the structural and the operational elements of the IMS are also detailed. Each individual factor has two parameters. It needs to select one parameter for each individual factor of the structural and operational elements to form an IMS. For example: Integration (lower degree integration), DSS (sequential execution method of refreshment and query function, sequential execution method of the tasks of the refreshment function, N number of task



of refreshment function means at least one task loading need to execute for the refreshment function), machine (low capacity), material (low volume of data, non-frequent change frequency of data), refreshment processing (non-continuous refreshment frequency method) and query processing (non-time constraint query method) can form an information manufacturing system. Further, Integration (the higher degree integration), DSS (sequential execution method of refreshment and query function, sequential execution method of the tasks of the refreshment function, N number of task of refreshment function means at least one task loading need to execute for the refreshment function), machine (low capacity), material (low volume of data, non-frequent change frequency of data), refreshment processing (non-continuous refreshment frequency method) and query processing (non-time constraint query method) can form another information manufacturing system and so on. Therefore, these parameters of the factors are the determinant of the heterogeneity of the IMS. The heterogeneous IMS is shown in Figure 3.8.



		Elements	Factors	Parameters of Factors
Heterogeneous Information Manufacturing System	Structural Elements	Integration	Degree of Integration	Low
				High
		Data Storage System	Execution Method of Refreshment & Query Function	Sequential
				Simultaneous
			Execution Method of Tasks of Refreshment Function	Sequential
				Simultaneous
			Execution of Number of Tasks for Refreshment Function	N
				N + 1
	Operational Elements	Machine	Capacity	Low
				High
		Material	Volume of Data	Low
				High
			Change Frequency of Data	Non-Frequent
				Frequent
		Refreshment Processing	Refreshment Frequency Method	Non-Continuous
				Continuous
		Query Processing	Query Method	Non-Time Constraint
				Time Constraint

**Figure 3.8: Determinants of Heterogeneity of IMS**

Moreover, the affected DQ dimensions for the DQ constraints such as continuity of available data, uniformity of available data and currency of available data was also discussed in this chapter. Chapter 4 will discuss the DQ assessment functions and the procedures for assessing DQ of heterogeneous IMS for showing poorness of DQ with timeliness.



## 4 Data Quality Assessment Functions & Procedures

---

### 4.1 Introduction

There is variation in the DQ of outbound data with inbound data in the IMS. There could be inbound DQ problems for typing mistakes, duplication of data from different sources, missing data or the malfunctions of the sources. Therefore, inbound DQ problems exist in the data storage system. In this case, outbound DQ will be affected for the DQ problems of inbound data. Further, time related DQ problems could occur in the IMS in the refreshment processing mechanism. Therefore, outbound DQ problems could be a combination of inbound DQ problems and time related DQ problems. This can be represented by the following formula.

$$\text{Outbound DQ Problem} = \text{Inbound DQ Problem} + \text{Time Related DQ Problem} \dots (4.1)$$

This formula can be explained more specifically. There could be incomplete DQ problems in the inbound data. Incomplete DQ problems could occur for time related factors. Therefore, incomplete outbound data will be the combination of the incomplete inbound DQ problems and incomplete DQ problems for time related factors. Similarly, there may be accuracy DQ problems in the inbound data. Furthermore, data could be obsolete for time related factors. Therefore, data may be inaccurate for time related factors. Hence, inaccurate outbound data could be the combination of the inaccurate inbound DQ problems and inaccurate DQ problems for time related factors.

There are many approaches for selecting outbound data with a query request from the DSS of the IMS. Attributes of the DSS are considered for selecting the outbound data



from data sets of the DSS. Some of the attributes of the DSS may need to be selected for a query request. Therefore, some data of each data set of the DSS will be retrieved for information support. Furthermore, some attributes of the DSS may need to be selected for a query request but a condition might be imposed. Therefore, only some data of some data sets will be retrieved for information support. Different approaches for selecting outbound data will now be further discussed.

Outbound data will be assessed after retrieving the outbound data with a query request in the IMS. Quality of outbound data will be compared with the corresponding inbound data for the change of DQ of outbound data. As a result, the retrieved outbound data and the corresponding inbound data will be assessed for DQ of outbound data. Therefore, the assessment function for assessing both inbound and outbound data will be the same. The only difference in the assessment will be the extra assessing function for quality of outbound data for time related issues. The objective DQ assessment functions are developed from the assessment functions of Capiello et al. (2003). Time related data quality assessment functions are written in this thesis for calculating the timeliness of data in IMS. The assessment functions used for assessing both inbound and outbound data are discussed.

## **4.2 Objective Data Quality Assessment Function**

DQ of the IMS is assessed by objective DQ dimensions. Therefore, assessment functions for objective DQ dimensions are developed. These assessment functions are used for measuring the DQ of the simulated IMS discussed in chapter 5.

**Completeness:** Completeness of time related data in the data storage system represents the ratio between the number of data stored in the DSS and the number of data that



should be stored in the DSS (Capiello et al., 2005). Completeness of data may be reduced due to time related issues. Completeness is decreased for the manipulation of data to the DSS and when it is unable to propagate the updated data to all DSS at the same time. It makes an effect to the uniformity of information constraint. Furthermore, if data is not available at the right time in the IMS for the refreshment processing period, it cannot produce complete information (Vrbsky, 1996). Completeness of data may be increased in the DSS of the IMS with time. The increase of completeness of data in the DSS can be calculated by considering completeness of data at the start time of the refresh period with completeness of data at the end time of the refresh period.

An objective measure of completeness of a data object is possible by adding up significant data values and comparing the result with the expected number of valid records. Completeness of data in the DSS can be measured by calculating the ratio between the number of complete values and the total number of values (Redman, 1996). It uses a benchmark database to make the comparison for calculating the completeness of data in the DSS. Therefore, the completeness dimension for each data item  $d_i$  is associated with a binary value as follows.

$$d_i \neq \text{null} \Rightarrow \text{completeness}(d_i) = 1$$

$$d_i = \text{null} \Rightarrow \text{completeness}(d_i) = 0$$

Completeness result of the data sets  $\pi_j$  will be calculated with the simple ratio form as follows.

$$\text{Completeness}(\pi_j) = \frac{\sum_{i=1}^N \text{Completeness}(d_i)}{N} \dots\dots\dots (4.2)$$

Where N is the total number of data item  $d_i$  in the data sets  $\pi_j$ .



Considering the completeness function of Capiello et al. (2003) the following function can be written:

$$Completeness (C_k) = 1 - \frac{\sum_{i=1}^M \sum_{j=1}^N Incompleteness (d_{ij})}{M \times N} \dots\dots\dots (4.3)$$

$C_k$  is the completeness of the data for each individual assessment of the data of query result for outbound DQ or the data of sources for inbound DQ. For example: multiple query results need to be assessed for measuring the outbound DQ. 'k' is 1 to n.  $d_{ij}$  is the data that should be in the particular location of the query result or the data sources. Data could be present or absent in the particular location of the query result or the data sources. The absent data is found by comparing the data of the query result or the data of the sources with the data of the benchmark database. Then, the absent data is counted to get the number of incomplete data for the first assessment, second assessment and so forth. Characters „i“ and „j“ indicate the attribute (column) and data set (row) respectively in the query result or the data sources. M and N are the total number of attributes and data sets respectively. Therefore,  $M \times N$  is the total number of data to be used for completeness measurement.

The completeness function can be modified to show the calculating process of completeness in detail for assessing both selected inbound and outbound data sets of the IMS for different approaches of selection of data. As completeness is measured by calculating the absent data of the selected inbound and outbound data sets, so, the modified assessment function for assessing both inbound and outbound data sets will be the same. Therefore, the modified assessment functions for completeness calculation for inbound and outbound data sets are given as:



$$\sum_{i=1}^M \sum_{j=1}^N Incompleteness (d_{ij}) = \sum_{i=1}^M |IDA_i| \dots \dots \dots (4.4)$$

$$|TD| = M \times N \dots \dots \dots (4.5)$$

$$Completeness (C_k) = 1 - \frac{\sum_{i=1}^M |IDA_i|}{|TD|} \dots \dots \dots (4.6)$$

Where, |IDA| = Number of Incomplete Data in Attribute, |TD| = Number of Total Data in Outbound Query Result for 100% Completeness or Number of Total Data in Inbound Data Source.

**Accuracy:** Accuracy is defined as the ratio between the number of correct values and the total number of values in the DSS. Time related data in the DSS can be inaccurate for obsolescence of data. This obsolescence of data can be measured by the timeliness dimension of data. Further, comparison of data will be based on a benchmark database. Therefore, accuracy dimension for each data item  $d_i$  is associated with a binary value as follows:

$$d_i \neq \overline{d_i} \Rightarrow Accuracy (d_i) = 0$$

$$d_i = \overline{d_i} \Rightarrow Accuracy (d_i) = 1$$

Where  $\overline{d_i}$  is the corresponding correct element contained in the benchmark database. Accuracy results of the data sets  $\pi_j$  will be calculated with the simple ratio form as follows.

$$Accuracy (\pi_j) = \frac{\sum_{i=1}^N Accuracy (d_i)}{N} \dots \dots \dots (4.7)$$

Where N is the total number of data items  $d_i$  in the data sets  $\pi_j$ .



Considering the accuracy assessment function of Capiello et al. (2003) the following accuracy assessment function for assessing the data or data sets of the IMS can be written:

$$Accuracy (A_k) = 1 - \frac{\sum_{i=1}^M \sum_{j=1}^N Inaccuracy (d_{ij})}{M \times N} \dots\dots\dots (4.8)$$

$A_k$  is the accuracy of the data for each individual assessment of the data of query result for outbound DQ or the data of sources for inbound DQ. For example: multiple query results need to be assessed for measuring the outbound DQ. 'k' is 1 to n.  $d_{ij}$  is the data of a particular location of the query result or the data sources. The data of the particular location can be inaccurate for not matching with the data of the benchmark database or matching with the data of the benchmark database but obsolete for time related issue. Then, each inaccurate piece of data is counted to get the number of inaccurate data for the first assessment, second assessment and so forth. „i“ and „j“ indicate the attribute (column) and data set (row) respectively in the query result or the data sources. M and N are the total number of attributes and data sets respectively. Therefore,  $M \times N$  is the total number of data to be used for accuracy measurement.

One reason for inaccuracy of both inbound and outbound data is different in a time-oriented point of view. Inbound data will be inaccurate for syntactical, symmental or duplication of data. These types of inaccurate data will be propagated to the outbound data if inbound DQ problems are not corrected. Therefore, time related DQ problems will be included with these types of DQ problems for outbound data. Hence, assessment functions are used for measuring the DQ of inbound and outbound data. Assessing term of the assessment function of inbound data will be included for assessing the outbound data. On the other hand, assessing term of outbound data for



time related DQ factor will not be included in the assessment function of inbound data for measuring inbound DQ. Therefore, the modified assessment function for showing the calculating process of accuracy in detail for measuring the inbound and outbound DQ can be written in the following way:

$$\sum_{i=1}^M \sum_{j=1}^N InAccuracy(d_{ij}) = \sum_{i=1}^M |IDA_i| \dots \dots \dots (4.9)$$

$$|TD| = M \times N \dots \dots \dots (4.10)$$

$$Accuracy(A_k) = 1 - \frac{\sum_{i=1}^M |IDA_i|}{|TD|} \dots \dots \dots (4.11)$$

Where,  $|IDA|$  = Number of Inaccurate Data in Attribute,  $|TD|$  = Number of Total Data in Each Individual Outbound Query Result or Number of Total Data in Inbound Data Source.

**Consistency:** A definition for consistency is given in chapter 3. According to that definition, consistency is defined as the property of multiple data values that do not conflict with each other. Redman (1996) states that consistency may be defined as value consistency and representational consistency. Conflicts between or among data values are examined as value consistency. Inconsistency is verified when two or more values cannot be correct at the same time. Representational consistency ensures that data values are represented in the same correct format. Value or representational consistency problems occur when integrating data in DSS from different sources. Reasons for occurrences of some value consistency are given in Table 4.1.



**Table 4.1: Example of Consistency Problem**

Reason of Consistency Problem	Example
Use of upper case and lower case letters	Department of Computing or department of computing
Use of abbreviation and acronyms	Department of Computing, Dept. of Computing
Word order	Cervantes Saavendra Miguel de or Muguel de Cervantes Saavendra
Punctuation marks (e.g. hyphens, commas, semicolons, brackets, exclamation marks etc.)	Multimedia labrotory (mmlab) or Multimedia labrotory-mmlab
Errors: misspelling, typing or printing errors	Bill Clinton or Bill Klinton
Numbers	Area 61 or Area Sixty One
Extra words	Royal Yacht Club or Yacht Club
Different denominations and synonyms	Seismological Register Unit or Seismic Register Unit
Use of different language	Tribunal de cuentas (Spanish) Court of Auditor (English)

According to Capiello et al. (2003) inconsistency occurs with conflict of two or more values for the same object (e.g. one user but different address). Consistency problems could occur if the corresponding value of an attribute is not suited for the respected attribute (user is male but sex is F or user is an adult but age is 12). There could be other consistency problems of data for the collection of data from multiple sources. For outbound data, consistency problems may occur for the query request that is sent at the same time but returns a different query result. Therefore, there is conflict in the set of data where one set of data is correct and the other set is incorrect. Measurement of inconsistent data volume can be ensured by comparing the data set of outbound data that comes at the same time.



### 4.3 Timeliness of Data in IMS

According to Batini and Scannapieco (2006) and Wang et al. (1993) timeliness can be defined as currency and volatility dimensions. More specifically it can be written:

$$\text{Max } (0, 1 - \text{currency/volatility}) \dots\dots\dots (4.12)$$

According to Batini and Scannapieco (2006) currency is defined as,  $\text{Currency} = \text{Age} + (\text{Delivery Time} - \text{Input Time})$ , Where Age measures how old the data unit is when received, Delivery Time is the time when information product is delivered to the user and Input Time is the time when data unit is obtained. On the other side, volatility is defined as the length of time data remains valid (Batini and Scannapieco, 2006). Obsolescence of data can be measured by the timeliness of data.

#### 4.3.1 Volatility of Data in IMS

Timeliness can be calculated from the currency and volatility dimension. As the definition of volatility is the length of time data remains valid (Batini and Scannapieco, 2006). Therefore, volatility of data depends on the expiry time of each individual data of the DSS. Expiry time of data depends on the change frequency of data (i.e. frequently changing and long-term changing). Expiry time of long-term changing data will be long. On the other hand, expiry time of frequently changing data is short. For short expiry time, data could be obsolete if processing time or age of the data is longer than the expiry time of the data. Therefore, the formula for the volatility of data in the DSS of the IMS is:

$$\text{Volatility} = \text{Expiry Time} - \text{Start of Data Insertion Time} \dots\dots\dots (4.13)$$



Start of data insertion time means starting of insertion time of data from sources to the DSS of the IMS. Start of data insertion from the sources can be represented as SIT. Expiry time can be represented as ET. Expiry time indicates the limit of the validity of data. Hence, the expiry time of all data of the DSS can be defined in the following way:

$$\text{Expiry Time of DSS Data (ET)} = \{ ET_L (D_i) \dots \dots \dots ET_H (D_i) \} \dots \dots \dots (4.14)$$

Therefore, the volatility of the entire data of the DSS can be calculated by considering the following formula:

$$\text{Volatility of DSS Data} = \int_{SIT}^{ET} d_i dt \dots \dots \dots (4.15)$$

As DSS contain the various types of change frequency of data, so, expiry time of data of the DSS will start from  $ET_L$ , the lowest expiry time of the data in the DSS. On the other side,  $ET_H$  is the highest expiry time of the data in the DSS. Data of the DSS can be divided into two different sets for change frequency of data:

$$\text{Non-Frequently Changing Data Set (LD}_i\text{)} = \{ ld_1, ld_2, ld_3, \dots \dots \dots ld_n \} \dots \dots (4.16)$$

$$\text{Frequently Changing Data Set (FD}_i\text{)} = \{ fd_1, fd_2, fd_3, \dots \dots \dots fd_n \} \dots \dots (4.17)$$

The set of non-frequently changing data of the DSS can vary for different expiry times.

$$LD_1 = ( ld_1, ld_2, \dots \dots \dots, ld_5 ) = ET_H \dots \dots \dots (4.18)$$

$$LD_2 = ( ld_6, ld_7, \dots \dots \dots, ld_{10} ) = ET_H + 1 \dots \dots \dots (4.19)$$

.

$$LD_n = ( ld_{(n-4)}, ld_{(n-3)}, \dots \dots \dots, ld_n ) = ET_H + n \dots \dots \dots (4.20)$$



This non-frequently changing data in the DSS contains long, longer or longest expiry time. Therefore, the highest limit expiry time of data exists in the non-frequently changing data of the DSS.

Similarly, the set of frequently changing data of the DSS can be varied for the different expiry times.

$$FD_1 = (fd_1, fd_2, \dots, fd_5) = ET_L \dots \dots \dots (4.21)$$

$$FD_2 = (fd_6, fd_7, \dots, fd_{10}) = ET_L + 1 \dots \dots \dots (4.22)$$

.

$$FD_n = (fd_{(n-4)}, fd_{(n-3)}, \dots, fd_n) = ET_L + n \dots \dots \dots (4.23)$$

This frequently changing data in the DSS contains short, shorter and shortest expiry time. Therefore, lowest limit expiry time of data lies to the frequently changing data of the DSS.

#### 4.3.2 Currency of Data in IMS

The Currency dimension of data in the DSS of the IMS depends on the age, delivery time and input time. These parameters were discussed from a general point of view. In the IMS, these parameters are shown in Table 4.2.



**Table 4.2: Currency Parameters of IMS**

General Currency Parameter	IMS Currency Parameter	Notation
Age	Waiting Period + Refreshment Processing Period	W + Rpro
Delivery Time	Query Response Time	DT
Input Time	Insertion Time of Data in DSS	IT

**Age:** It can be calculated in the IMS by adding waiting period of data to the refreshment processing period of data. The waiting period is how long data waits in the source before the start of refreshment processing of data in the IMS for the insertion of data in the DSS. Rpro is calculated by adding the following parameters.

**Table 4.3: Refreshment Processing Period Parameters**

Refreshment Processing Period Parameter	Description	Notation
Loading Period	Time needed for loading data in the DSS	L
Indexing Period	Time needed for indexing data in the DSS	Ix
Propagation Delay	Time needed for propagating data from one DSS to another DSS.	P

Therefore, the refreshment processing period of data in the DSS can be calculated in the following way:

$$Rpro = L + Ix + P \dots\dots\dots(4.24)$$

Duration of loading period, indexing period and propagation delay may vary for the manipulation frequency of data. Manipulation of data in the DSS of the IMS is done in the following three ways.

**Insertion:** Some of the sources create a new set of data that increase the storage of data and change the scenario of data storage. Insert frequency describes the number of the



insert operation that occurred from the sources to the data storage system in the period of refreshment. It is denoted by If.

**Deletion:** Some of the sources delete an existing data or data set. As a result, data is decreased in the data storage system (DSS). Therefore, delete frequency describes the number of delete operation that occurred from the sources to the data storage system in the period of refreshment. It is denoted by Df.

**Updating:** Some of the sources update an existing data or data set. As a result, the row does not increase in size, rather the contents of some data change. The update process first deletes the existing data or data set and then inserts the new data or data set to complete the process. Hence, the updating process can be shown as:

$$\text{Updating: } Delete_{old} \longrightarrow Insert_{new}$$

Therefore, for updating a data unit or data set in the data storage system, both delete and insert operation execute one after another respectively. Hence, update frequency describes the number of the update operation that occurs from the sources to the data storage system in the period of refreshment. It is denoted by Uf.

The IMS integrates multiple sources with the data storage system. These sources will manipulate data in the data storage system. There could have several conditions of manipulations for manipulating data in the DSS from among the sources. These conditions are given in Table 4.4. In this table, VDM indicates the volume of data manipulation.



**Table 4.4: Probable Simultaneous Manipulation Operations in IMS from Multiple Sources**

Manipulation Operation	Description	Formula
Insert Only	All sources of the system only execute the insert operation	$\forall S_m \in S (S_m = If) \Rightarrow VDM = \sum_{If=1}^p If$
Delete Only	All sources of the system only execute the delete operation	$\forall S_n \in S (S_n = Df) \Rightarrow VDM = \sum_{Df=1}^p Df$
Update Only	All sources of the system only execute the update operation	$\forall S_o \in S (S_o = Uf) \Rightarrow VDM = \sum_{Uf=1}^p Uf$
(Insert + Delete) Only	Some sources of the system will execute the insert operation and some will execute delete operation	$\exists S_m \exists S_n \in S (S_m = If, S_n = Df) \Rightarrow VDM = \left( \sum_{If=1}^p If + \sum_{Df=1}^p Df \right)$
(Insert + Update) Only	Some sources of the system will execute the insert operation and some will execute the update operation	$\exists S_m \exists S_o \in S (S_m = If, S_o = Uf) \Rightarrow VDM = \left( \sum_{If=1}^p If + \sum_{Uf=1}^p Uf \right)$
(Update + Delete) Only	Some sources of the system will execute the update operation and some will execute delete operation	$\exists S_n \exists S_o \in S (S_n = Df, S_o = Uf) \Rightarrow VDM = \left( \sum_{Df=1}^p Df + \sum_{Uf=1}^p Uf \right)$
(Insert + Delete + Update) Only	Some sources of the system will execute the insert operation and some will execute delete operation and some others will execute the update operation	$\exists S_m \exists S_n \exists S_o \in S (S_m = If, S_n = Df, S_o = Uf) \Rightarrow VDM = \left( \sum_{If=1}^p If + \sum_{Df=1}^p Df + \sum_{Uf=1}^p Uf \right)$



**Insertion Time:** The time it takes to insert a new data item. This value includes the time it takes to find the correct place to insert the new data item as well as the time it takes to update the index structure.

**Deletion Time:** The time it takes to delete an existing data item. This value includes the time it takes to find the item to be deleted as well as the time it takes to update the index structure.

**Update Time:** The time it takes to update a data item. This value includes the time it takes to find the item to be deleted first plus the time it takes to insert the new data item. It also needs time to update the index structure, which will also be included in calculating the updating time.

Among the manipulation process (Insertion, Deletion and Updating), data have to be loaded (insert, update) or unloaded (delete, update) at the beginning. Therefore, the index updating function will be executed. To make data available in the DSS, execution of loading or unloading and indexing function is mandatory. As a result, data will not be available until loading and indexing period is completed. Moreover, for the replication DSS (cluster DSS) or the replication type DSS, propagation delay will be added to be available of data in each DSS. Hence, the formula for the calculation of loading period, indexing period and propagation delay are given below:

Loading period for a data or set of data can be calculated by subtracting end of loading time for a data or set of data from the start of the loading time of that same data or set of data. Similarly, a de-loading period of a data or set of data can be calculated by subtracting the start of de-loading time of a data or a set of data from the end of de-loading time of that same data or set of data. Therefore, it can be written:



$$\text{Loading Period for a Set of Data (L)} = EL - SL \dots\dots\dots(4.25)$$

$$\text{De-Loading Period for a Set of Data (DL)} = EDL - SDL \dots\dots\dots(4.26)$$

Where, EL = End of Loading Time, SL = Start of Loading Time and EDL = End of De-Loading Time, SDL = Start of De-Loading Time.

Therefore, the total loading and de-loading period (TL) for the sets of manipulating data is:

$$TL = ( \sum_{If=1}^n If \times L + \sum_{Df=1}^n Df \times DL + \sum_{Uf=1}^n Uf \times (DL + L) ) \dots\dots(4.27)$$

„If“ the insert frequency that calculates the number of insertions in the DSS. Total loading period of insertion can be calculated by multiplying total number of insertions by the loading period for a set of data. „Df“ is the delete frequency that calculates the number of deletions from the DSS. Total de-loading period for deletion can be calculated by multiplying total number of deletions by the de-loading period for a set of data. „Uf“ is the update frequency that calculates the number of updating a set of data in the DSS. As, updating process of a set of data is done by executing the deletion and insertion function respectively, both the de-loading and loading period of a set of data have to be considered for calculating the update period of a set of data. Hence, total updating period can be calculated by multiplying the total number of updating sets of data by the de-loading and loading period of a set of data.

Indexing period for a data or a set of data can be calculated by subtracting end of index time for a data or a set of data from the start of the index time of that data or set of data. Therefore, it can be written:



$$\text{Indexing Period for a Set of Data (Ix)} = \text{EIx} - \text{Six} \dots \dots \dots (4.28)$$

Where, EIx = End of Indexing Time and Six = Start of Indexing Time

Therefore, the total indexing period (TIx) for the sets of manipulating data is:

$$\text{TIx} = (\sum_{If=1}^n If \times Ix + \sum_{Df=1}^n Df \times Ix + \sum_{Uf=1}^n Uf \times Ix) \dots \dots \dots (4.29)$$

„If“, „Df“ and „Uf“ indicates the insert, delete and update frequency respectively. The index structure of the DSS updates for insertion, deletion and updating of data. Therefore, the indexing period for inserting data can be calculated by multiplying the insert frequency by the indexing period for a set of data. Similarly, the indexing period for deleted data can be calculated by multiplying the delete frequency by the indexing period for a set of data. Indexing period calculation of update frequency is the same as the sum of the insert and delete frequency. If the sources of the system executes among manipulation (insert, update, delete), then the indexing period will be calculated by adding the re-organizing period of index for insertion, deletion and updating of data.

Propagation delay for a data or a set of data can be calculated by subtracting end of the propagation time for a data or a set of data from the start of the propagation time of that data or set of data. Therefore, it can be written:

$$\text{Propagation Delay for a Set of Data (P)} = \text{EP} - \text{SP} \dots \dots \dots (4.30)$$

Where, EP = End of Propagation Time and SP = Start of Propagation Time

Therefore, total propagation delay (TP) for the sets of manipulating data is:

$$\text{TP} = (\sum_{If=1}^n If \times P + \sum_{Df=1}^n Df \times P + \sum_{Uf=1}^n Uf \times P) \dots \dots \dots (4.31)$$



Data is propagated in the replication DSS (cluster DSS) or replication type DSS. Therefore, there is a propagation delay in updating the data in each DSS. In the formula, „If“ indicates the propagated data for insert frequency; „Df“ indicates the propagated data for deleting frequency and „Uf“ indicates the propagated data for update frequency. The propagation delay for insert frequency can be calculated by multiplying the propagation delay of a set of data with the insert frequency of data in the DSS. Similarly, the propagation delay for deletion and updating of data in the DSS can be calculated by multiplying the delete frequency and update frequency by the propagation delay for a set of data respectively. If among manipulation (insert, delete, update) of data in DSS executes simultaneously from the sources, then, propagation delay will be calculated by adding the propagation delay for each type of manipulation frequency.

**Input Time:** To be available data have to be inserted in the DSS of the IMS. Data insertion will be complete if refreshment processing of data in the DSS is carried out. Therefore, the end of the refreshment processing period will be the input time of the data.

**Delivery Time:** It is defined in the DSS by query response time. The query response time of DSS means, what time the query request of a user query is responded in the DSS.

#### **4.3.2.1 Single DSS Oriented IMS**

Currency calculation of data in the DSS depends on age, input time and query response time. Age can be calculated by the refreshment processing period and waiting period of the data in the source. Refreshment processing period of data in the single DSS rely on

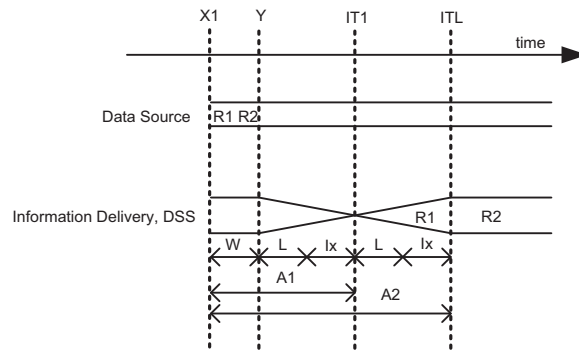


the loading and indexing period of data available in the DSS. Tasks of the refreshment process execute sequentially in the single DSS. Refreshment processing period depends on the number of tasks and the execution method of the tasks of the refreshment process. Data insertion or input time of data in the DSS is controlled by the refreshment processing period. Query response time is the delivery time of data from the DSS. Waiting period of data depends on the staying period of data in the source before the start of the refreshment process. Therefore, waiting period of data can vary. Currency formula for the single DSS can be written as:

$$\begin{aligned}
 \text{Currency } (C) &= A + (DT - IT) \\
 &= (W + R_{pro}) + (DT - IT) \\
 &= (W + n(L + I_x)) + (DT - IT) \dots \dots \dots (4.32)
 \end{aligned}$$

Where A is the age of the data, DT and IT is the delivery time and insertion time of data in DSS respectively. W, L and I<sub>x</sub> represent the waiting period, loading period and the indexing period respectively for calculating the age of the data of the single DSS. „n“ is the number of times loading and indexing tasks of the refreshment function execute to make each data of the source available in the DSS. „n“ is greater than 0 (n>0). To make the formula more understandable, Figure 4.1 is given below:





**Figure 4.1: Currency of the Data in Single DSS Oriented IMS**

In the figure 4.1, when there is no cross line, it means that the refreshment processing function is not working in DSS in that period. On the other hand, when there is a cross line, it means that the refreshment processing function is working in DSS in that period.

It is also shown in Figure 4.1 that there are two sets of data (R1 and R2) in the data source. The waiting period is there for the time delay before the start of data insertion to DSS or the start of refreshment processing to make the source data available in the DSS. R1 is inserted at IT1 after the completion of loading and indexing tasks. R2 is inserted at ITL after the completion of another loading and indexing task. This data insertion process will continue in the same way if more data was in the source. The refreshment process of R1 is completed before the refreshment process of R2. Therefore, age of the data of R1 is calculated by adding loading and indexing period to the waiting period. On the other hand, age of the data of R2 is calculated by adding loading and indexing period of both R1 and R2 to the waiting period. For this reason, „n“ is used in the formula to calculate the age of each data of the source inserted in DSS. R1 can be delivered at any time after IT1. Therefore, currency of the R1 is calculated from age, input time and delivery time of R1. Currency of R2 and if there would be more data can be calculated in the same way of R1.



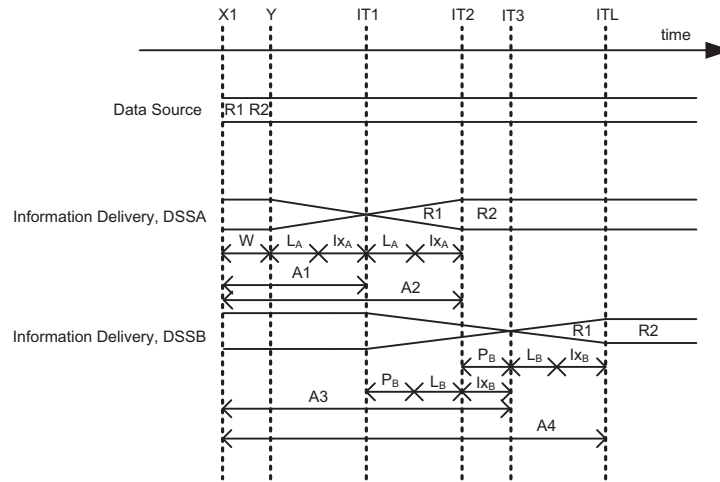
#### 4.3.2.2 Cluster DSS Oriented IMS

Multiple DSS work in the cluster DSS oriented IMS. Therefore, data can be manipulated in any of the DSS of the cluster. Manipulated data are propagated to other DSS of the cluster. Hence, there is a propagation delay for updating among DSS of the cluster. Therefore, refreshment processing period will be calculated by adding propagation delay with the indexing and loading period. Data insertion time or input time of DSS will be controlled by this refreshment processing time. Waiting period of data depends on the staying period of data in the source before the start of the refreshment process. Waiting period of data can vary. Age is calculated from the refreshment processing period and the waiting period of the data in the source. Query response time works in the same way as it works in the single DSS. Query response time is the delivery time of data. Therefore, currency of the data in the cluster DSS is:

$$\begin{aligned}
 \text{Currency (C)} &= A + (DT - IT) \\
 &= (W + R_{pro}) + (DT - IT) \\
 &= (W + n(L + I_x) + mP) + (DT - IT) \dots \dots \dots (4.33)
 \end{aligned}$$

Where A is the age of the data, DT and IT is the delivery time and insertion time of data in DSS respectively. W, L, I<sub>x</sub> and P represent the waiting period, loading period, indexing period and propagation delay respectively for calculating the age of the data of the cluster DSS. „n“ is the number of times loading and indexing tasks of the refreshment function execute to make each data of the source available in each DSS of the cluster. „m“ is the number of times data is propagated to the DSS of the cluster. „n“ is greater than 0 (n>0). Further, „m“ is equal or greater than 0 (m>=0). To make the formula more understandable, Figure 4.2 is given below:





**Figure 4.2: Currency of the Data in Cluster DSS Oriented IMS**

It is shown in Figure 4.2 that there are two sets of data (R1 and R2) in the data source. The waiting period is there for the time delay before the start of data insertion to DSS or the start of refreshment processing to make the source data available in the DSS. Data needs to be inserted into the multiple DSS of the cluster. Therefore, refreshment process of data will be completed after insertion of data into the last DSS of the cluster. The last DSS in the figure is DSSB. R1 is inserted at IT1 in DSSA after the completion of loading and indexing tasks in DSSA. This R1 is inserted at IT3 in DSSB after the completion of loading and indexing tasks in both DSSA and DSSB and propagation of data task for propagating data from DSSA to DSSB. R2 is inserted at IT2 after the completion of another loading and indexing tasks in DSSA. R2 is inserted at ITL after the completion of loading and indexing tasks of R1 and R2 in DSSA and propagation task for inserting data from DSSA to DSSB and then loading and indexing tasks for R2 in DSSB. This data insertion process will continue in the same way if more data was in the source. The refreshment process of R1 is completed before the refreshment process of R2. Therefore, age of the R1 for DSSA is calculated by adding loading and indexing period of R1 to the waiting period in DSSA. Age of the R2 for DSSA is calculated by



adding loading and indexing period of both R1 and R2 to the waiting period in DSSA. Further, age of the R1 for DSSB is calculated by adding loading and indexing period of both DSSA and DSSB and propagation period of DSSB to the waiting period. On the other hand, age of the R2 for DSSB is calculated by adding loading and indexing period of both R1 and R2 in DSSA and loading, indexing and propagation period of DSSB to the waiting period. „n“ is used in the formula because loading and indexing tasks will be executed in each DSS of the cluster. Further, in the first DSS of the cluster, data will be loaded and indexed one after another. Therefore, „n“ number of loading and indexing tasks and „m“ number of propagation tasks calculate the age of each data of the source inserted in DSS. The number of propagation tasks depends on the number of DSSs are in the cluster. R1 can be delivered at any time after IT1 and IT3 from DSSA and DSSB respectively. Therefore, currency of the R1 is calculated from the age, input time and delivery time of R1. Currency of R2 and if there would be more data can be calculated in the same way of R1.

#### **4.3.2.3 2-DSS Oriented IMS**

In 2-DSS, one DSS loads the data temporarily quick provision of information support and another DSS holds the data permanently. Refreshment processing period of data will be the loading period. Waiting period of data depends on the staying period of data in the source before the start of the refreshment process. Therefore, waiting period of data can vary. Age of the data will be the refreshment processing period if there is no waiting period. If there is a waiting period of data, the age of the data will be the waiting and refreshment processing period. The permanent DSS of 2-DSS works as a single DSS when data moves from temporary DSS to permanent DSS. Data of temporary table (DSS) are loaded and indexed into the permanent table (DSS). When,

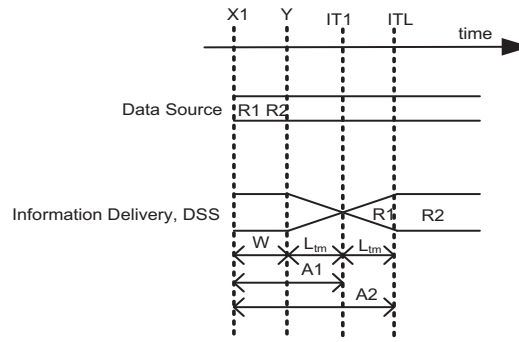


data is moved from the temporary DSS to the permanent DSS, both DSS can go offline or be online. The temporary DSS can go offline but permanent DSS can provide information support at the refreshment processing period of data from temporary DSS to permanent DSS. As a result, the refreshment processing period of data in temporary DSS is the waiting period of the data of permanent DSS. Hence, waiting period equals the loading period of data of temporary DSS. Waiting period of each data of the temporary DSS is not the same. Waiting period of the data that loads in the temporary DSS at the very beginning and the data that loads in the temporary DSS at the end varies. Age of the data is then calculated by adding the waiting period of data in the temporary DSS with the loading and indexing period of data in the permanent DSS. Henceforth, the currency of the data of 2-DSS has to be calculated for two different states. One state is the temporary DSS state and the other the permanent DSS state. Query request executes in both temporary and permanent DSS for information support. Therefore, there is a query response time. This query response time is the delivery time of data. Currency of data for the temporary DSS state is:

$$\begin{aligned}
 \text{Currency } (C) &= A + (DT - IT) \\
 &= (W + R_{pro}) + (DT - IT) \\
 &= (W + nL_{tm}) + (DT - IT) \dots\dots\dots (4.34)
 \end{aligned}$$

Where A is the age of the data, DT and IT is the delivery time and insertion time of the data in the DSS respectively.  $L_{tm}$  represents the loading period of data in temporary DSS. „n“ is the number of times loading task of the refreshment function execute to make each data of the source available in the DSS. „n“ is greater than 0 ( $n > 0$ ). To make the formula more understandable, Figure 4.3 is given below:





**Figure 4.3: Currency of the Data in 2-DSS (Temporary) Oriented IMS**

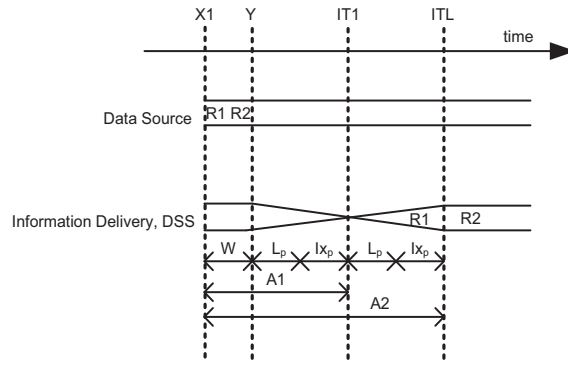
It is shown in Figure 4.3 that there are two sets of data (R1 and R2) in the data source. The waiting period is there for the time delay before the start of data insertion to DSS or the start of refreshment processing to make the source data available in the DSS. R1 is inserted at IT1 after the completion of loading task. R2 is inserted at ITL after the completion of another loading task. This data insertion process will continue in the same way if more data was in the source. The refreshment process of R1 is completed before the refreshment process of R2. Therefore, age of the R1 is calculated by adding loading period to the waiting period. On the other hand, age of the R2 is calculated by adding loading period of both R1 and R2 to the waiting period. For this reason, „n“ is used in the formula to calculate the age of each data of the source inserted in DSS. R1 can be delivered at any time after IT1. Therefore, currency of the R1 is calculated from the age, input time and delivery time of R1. Currency of R2 and if there would be more data can be calculated in the same way of R1.



Now, Currency of data for the permanent DSS state is:

$$\begin{aligned}
 \text{Currency } (C) &= A + (DT - IT) \\
 &= (W + R_{pro}) + (DT - IT) \\
 &= (nL_{tm} + n(L_p + Ix_p)) + (DT - IT) \dots \dots \dots (4.35)
 \end{aligned}$$

Where A is the age of the data, DT and IT is the delivery time and insertion time of the data in the DSS respectively.  $L_p$ ,  $Ix_p$  and  $L_{tm}$  represent the loading period of data in permanent DSS, indexing period of data in permanent DSS and the loading period of data in temporary DSS respectively for calculating the age of the data of the 2-DSS. „n“ is the number of times loading and indexing tasks of the refreshment function execute to make each data of the source available in the DSS. „n“ is greater than 0 ( $n > 0$ ). To make the formula more understandable, Figure 4.4 is given below:



**Figure 4.4: Currency of the Data in 2-DSS (Permanent) Oriented IMS**

It is shown in Figure 4.4 that there are two sets of data (R1 and R2) in the data source. The waiting period is there for the time delay before the start of data insertion to DSS or the start of refreshment processing to make the source data available in the DSS. R1 is inserted at IT1 after the completion of loading and indexing tasks. R2 is inserted at ITL after the completion of another loading and indexing task. This data insertion



process will continue in the same way if more data was in the source. The refreshment process of R1 is completed before the refreshment process of R2. Therefore, age of the R1 is calculated by adding loading and indexing period to the waiting period. On the other hand, age of the R2 is calculated by adding loading and indexing period of both R1 and R2 to the waiting period. For this reason, „n“ is used in the formula to calculate the age of each data of the source inserted in DSS. R1 can be delivered at any time after IT1. Therefore, currency of the R1 is calculated from the age, input time and delivery time of R1. Currency of R2 and if there would be more data can be calculated in the same way of R1.

#### **4.3.2.4 3-DSS Oriented IMS**

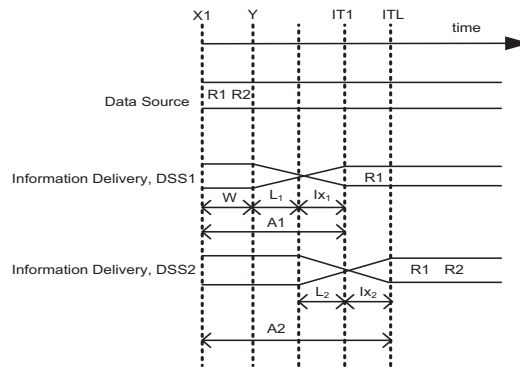
Functionalities of the DSS and the tasks of the refreshment functions (loading, indexing, query processing) work simultaneously in the 3-DSS of the IMS. Therefore, it needs less time to execute the functionalities than in the single DSS. However, data will not be available in the DSS until loading and indexing of data are complete. As three functionalities execute in the individual DSS and index structure exists in each individual DSS, so, loaded data will not be available until the indexing is done. Therefore, the refreshment processing period of the data unit will be the summation of the loading and the indexing period of data. Waiting period of data depends on the staying period of data in the source before the start of the refreshment process. Therefore, waiting period of data can vary. Hence, the age of the data in 3-DSS will be calculated with the refreshment processing period and the waiting period. Query response time and data insertion or input time of data in DSS work as it does for other DSS (Cluster DSS, Single DSS etc.). Therefore, currency of the data in 3-DSS can be calculated in the following way:



$$\begin{aligned}
\text{Currency (C)} &= A + (DT - IT) \\
&= (W + R_{pro}) + (DT - IT) \\
&= (W + (L + nIx)) + (DT - IT) \dots \dots \dots (4.36)
\end{aligned}$$

Where A is the age of the data, DT and IT is the delivery time and insertion time of data in DSS respectively. W, L and Ix represent the waiting period, loading period and indexing period respectively for calculating the age of the data of the 3-DSS. At the beginning of the 3-DSS, only loading of data will be done in one DSS. No indexing will be done. Therefore, L is added before the nIx in the formula. „n“times of indexing period are used in the currency formula because loading and indexing tasks are executed simultaneously in two different DSS. Therefore, loading and indexing of data are done in the same period of time. However, indexing period is taken for „n“times but not the loading period because indexing of the data takes more time than the loading of a data. As a result, more than one data is loaded in one DSS at the time of indexing of data in another DSS. Further, after the insertion of the last source data only indexing task will be executed in one DSS. No loading of data will be done. „n“is the number of times indexing task of the refreshment function execute to make each data of the source available in the DSS. „n“ is greater than 0 ( $n > 0$ ). To make the formula more understandable, Figure 4.5 is given below:





**Figure 4.5: Currency of the Data in 3-DSS Oriented IMS**

It is shown in Figure 4.5 that there are two sets of data (R1 and R2) in the data source. The waiting period is there for the time delay before the start of data insertion to DSS or the start of refreshment processing to make the source data available in the DSS. R1 is inserted at IT1 after the completion of loading and indexing tasks. R2 is inserted at ITL after the completion of another loading and indexing task. This data insertion process will continue in the same way if more data was in the source. The refreshment process of R1 is completed in DSS1 before the refreshment process of R1 and R2 in DSS2. Therefore, age of the R1 of DSS1 is calculated by adding loading and indexing period to the waiting period. On the other hand, age of the R1 and R2 of DSS2 is calculated by adding loading and indexing period of R1 of DSS1 and indexing period of DSS2 to the waiting period. The loading period of R1 and R2 in DSS2 is not added because the loading task of R1 and R2 of DSS2 is executed in the same period as the indexing task of R1 in DSS1. For this reason loading period of first inserted source data and „n“ number of indexing period are used in the formula to calculate the age of each data of the source inserted in DSS. R1 can be delivered at any time after IT1. Therefore, currency of the R1 in DSS1 is calculated from the age, input time and



delivery time of R1. Currency of R1 and R2 in DSS2 and if there would be more data can be calculated in the same way of R1 of DSS1.

#### **4.4 Comparison of Inbound & Outbound Data Quality**

Inbound data coming from multiple sources are stored in the data storage system. Inbound data will be assessed by the inbound DQ assessment function for the measurement of the quality of inbound data. An assessment will be conducted just before insertion of data in the data storage system. Similarly, outbound data will be assessed by the outbound DQ assessment function for measurement of the quality of outbound data. Assessment of outbound data will be conducted just after the execution of the query request for outbound data. A comparison of both inbound and outbound data will be compared with a benchmark database discussed in chapter 5. By comparing both inbound and outbound data with the benchmark database, we will get the DQ information of both inbound and outbound data. Hence, outbound DQ will be compared with inbound DQ to find the change of DQ in the IMS.

##### **4.4.1 Comparison Approach of Inbound and Outbound Data Quality**

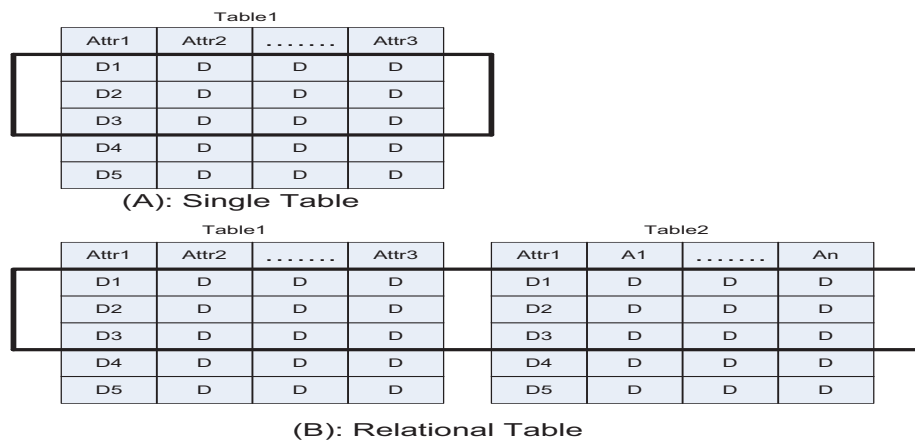
Comparison of two items or multiple items can show the best item of two or more items. It can also show how good one item is compared with the other item(s). In the IMS, the comparison function will be used to compare the DQ of inbound and outbound data. Therefore, comparison approaches can be used for comparing the quality of inbound data with the quality of outbound data. Comparison approaches are designed by considering the possible categories query request (Silberschatz et al., 1997). These comparison approaches are now discussed.



#### 4.4.1.1 Random Data Selection Approach

In this approach, a part of the database data will be selected as outbound data for the query request. Quality of outbound data will be measured using the DQ assessment function. Similarly, assessing part of outbound data of inbound database will be assessed for the quality measurement of inbound data. Finally, quality of outbound data is compared with the quality of inbound data. Random data selection approaches can be carried out in multiple ways.

**Random-Horizontal Data Selection Approach:** Data are stored in a logical data storage system such as a table. Identification of data (attributes) are organised vertically. Rows of the table that hold the data for each attribute are laid down horizontally. In the random-horizontal data selection approach, each attribute of the logical DSS (tables) will be selected, but each data set of those logical DSS (tables) won't select. This is shown in Figure 4.6.



**Figure 4.6: Random Horizontal Data Selection Approach**

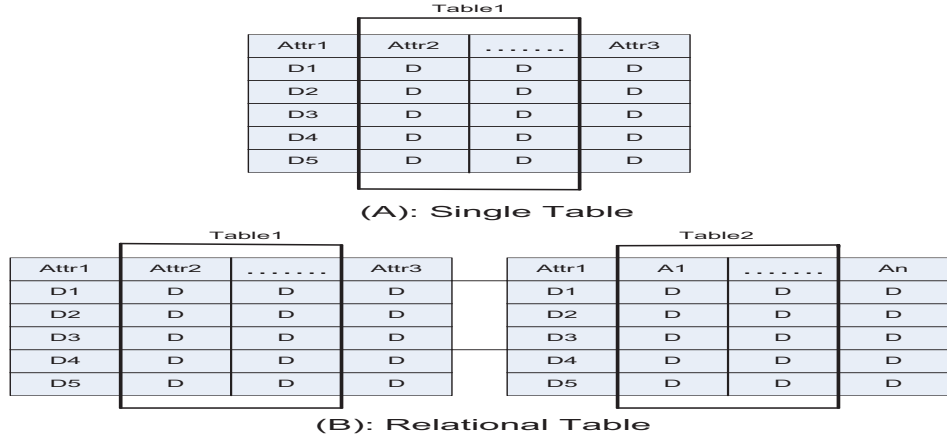
Figure 4.6 depicts the random-horizontal data selection approach according to the following query expression.



Single Table:  $\Pi_{(Attr1, Attr2, Attr3, \dots, Attrn)} (\sigma_{Attr1 < D4} (Table1))$

Relational Table:  $\Pi_{(Attr1, Attr2, Attr3, \dots, Attrn)} (\sigma_{Attr1 < D4} (Table1)) \cup \Pi_{(Attr1, Attr2, Attr3, \dots, Attrn)} (\sigma_{Attr1 < D4} (Table2))$

**Random-Vertical Data Selection Approach:** Each datum of some attributes of the table or tables will be selected at random-vertical data selection approach. Therefore, a selection of attributes will be the subset of attributes of a table or tables. Similarly, selection of data for the particular attributes will be the subset of the data of a table or tables. In random-vertical data selection approach, selected data cannot be the aggregation of sets of data. Rather, the selected data will be the subset of the data of each individual table. Therefore, vertically selected data will be the aggregation of the subset of data of each individual data set of a table.



**Figure 4.7: Random Vertical Data Selection Approach**

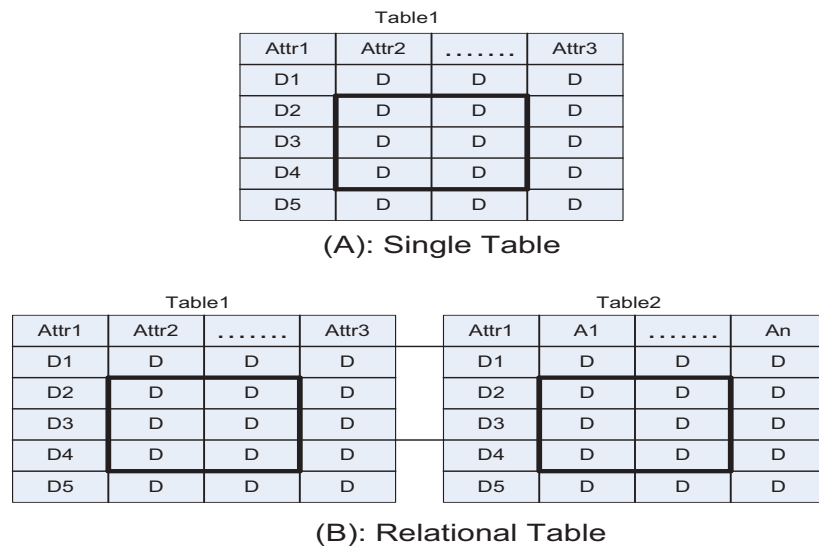
Figure 4.7 depicts the random-vertical data selection approach according to the following query expression.

Single Table:  $\Pi_{(Attr2, Attr3)} (Table1)$

Relational Table:  $\Pi_{(Attr2, Attr3)} (Table1) \cup \Pi_{(A2, A3)} (Table2)$



**Partially Mixed (H+V) Random Data Selection Approach:** In this approach, both the selected attributes and the selected data of the data sets are the subset of the set of attributes of a table or tables and the subset of the individual data set of a data storage system or table or tables respectively. Data of a DSS are located vertically with respect to the attributes of the DSS. The data are located horizontally with respect to the rows. In this approach, the selected data will be located in the crossing point of the vertically selected attributes and the selected data of the horizontally located rows. Therefore, this approach can be recognized as partially mixed (H+V) random data selection approach.



**Figure 4.8: Partially Mixed (H+V) Random Data Selection Approach**

To make this approach more understandable, Figure 4.8 depicts the random partial horizontal-vertical data selection approach according to the following query expression.

Single Table:  $\Pi_{(Attr2, Attr3)} ( \sigma_{(Attr1 > D1 \wedge Attr1 < D4)} (Table1) )$

Relational Table:  $\Pi_{(Attr2, Attr3)} ( \sigma_{(Attr1 > D1 \wedge Attr1 < D4)} (Table1) ) \cup \Pi_{(Attr2, Attr3)} ( \sigma_{(Attr1 > D1 \wedge Attr1 < D4)} (Table2) )$



These approaches are presented to understand the types of query requests are sent in IMS to get the outbound data in the real world. DQ of heterogenous IMS will be measured. Therefore, Similar part of an approach will be used for each experiment of IMS in this research.

## **4.5 Summary**

DQ assessment functions for objective DQ dimensions are developed in this chapter for assessing DQ of the heterogeneous IMS. Timeliness DQ function for each individual DSS of the heterogeneous IMS are written from the general timeliness DQ dimension. Procedure of the assessment of DQ of the heterogeneous IMS is given in this chapter. Chapter 5 will show the effect on DQ or change of DQ with timeliness in the IMS both analytically and experimentally using the procedure and assessment functions described in this chapter.



## **5 Diversification of Data Quality with Timeliness in Heterogeneous IMS**

---

### **5.1 Introduction**

DQ of the IMS is changed with the change of timeliness. It is seen that timeliness is calculated by the currency and volatility. Therefore, timeliness of data is measured by the currency and volatility. This currency and volatility can make an impact on the timeliness of data. Currency of data depends on the age, the insertion time of the data in the DSS (Input Time) and the query response time (Delivery Time). Furthermore, volatility is the validity of data. It depends on the length between the expiry time of the data and the insertion time of the data in the system. Variations of the refreshment processing period can be done for the usage of the DSS in the IMS and the execution process of the tasks of the functionalities of the data storage system. Therefore, variations of the refreshment processing period of data can effect on DQ constraints such as continuity of available data, current available data and uniform available data in the IMS. These constraints of DQ in turn can affect completeness, accuracy and consistency DQ dimensions.

Timeliness of information is whether the set of data comes at the right time or not. According to this definition, a set of data is to come at the right time. Otherwise, the data set will have expired. Therefore, query request responding time (delivery time) is important for the DQ of the set of data (information) in the IMS. If the query request does not respond at the right time, there will be a DQ problem of obsolescence of data in the retrieved data set for the query request. If the query request responds at the right



time, however, there could be a completeness DQ problem in the retrieved data set for the query request. The set of data will come from the IMS. If the data set of the IMS is current, each individual data from the IMS could be valid from a data timeliness point of view. On the other hand, the data of the IMS could be invalid or be of poor quality from the information timeliness point of view. For an example, let,  $d_1$ ,  $d_2$ ,  $d_3$ ,  $d_4$  and  $d_5$  are five data in a data set for a query request. Expiry time of each individual data is 4:20 and the obligatory right time (expiry time) for this data set is 4:15. Now, if the system fails to deliver this requested query data set within the obligatory right time (4:15), the data set will be poor quality for the obsolescence. On the other hand, if the data set does not have the obligatory right time, then, expiry time of the data set will be the expiry time of the individual data (4:20). Therefore, volatility of the data set will be varied for whether there is an obligatory right time constraint or not. Furthermore, currency of the set of data will vary for the variations of the responding time of the query request. As a result, timeliness of the set of data will vary for the variation of the currency value and the volatility of the set of data. As data is processed to get the set of data for a query request, therefore, DQ (accuracy, completeness and consistency) could be changed with the change of the information timeliness.

It is seen in chapter 3; IMSs are constructed by machine, material and different methods. According to Wang et al. (2008) machine, material and method play the DQ regulating role in the IMS. This chapter will show the change and improvement of DQ with timelines for machine, material and methods in IMS.

Combined research methodology can improve the quality of the research (Kaplan and Duchon, 1988). Therefore, two research methodologies have been used for identifying the cause for the change and improvement of DQ with timeliness in IMS. The



methodologies used for this thesis are math modelling (theoretical or analytical), and simulation.

Math modelling models the real world and states the result as mathematical equations. All the dependent and independent variables are included in this model. This methodology is considered to be the highest order of methodology by many researchers (Jenkins, 1985). Math modelling is used for showing the affected DQ dimension with respect of timeliness from a theoretical point of view.

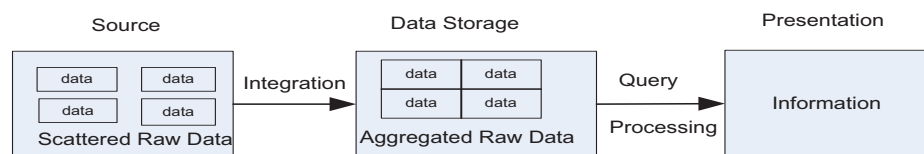
Simulation can be defined as mirroring a segment of the real world or copying the behaviour of a system. Real world IMS has been simulated for measuring the DQ with timeliness following the theoretical DQ assessment. The DQ measurements with timeliness of IMS are shown both theoretically and experimentally by the simulation based IMS.

## **5.2 Diversification of Data Quality with Timeliness in IMS**

The sources of the IMS could be either the real world state or multiple environmental sources (finance, engineering etc.) or in operational databases for the larger organization. Extracted or inserted raw data can be defined as scattered raw data as they come from different sources. This scattered data may not be in the same data format and the source format could also be different. Furthermore, scattered raw data could be of poor quality from the incompleteness, inaccuracy, inconsistency point of view. Therefore, scattered raw data from sources will be transformed and cleaned before loaded into the data storage system. The loaded data coming from various sources to data storage block can be defined as the inbound data. Scattered raw data from multiple sources then integrate the raw data and store the data in the data storage system as



aggregated raw data. These aggregated raw data could be a combination of stable, long-term changing and frequently changing data. The data storage system prepares the aggregated raw data for information with the refreshment process. Each manipulation of data in the sources refreshes the data storage system to align the manipulated data in the DSS of the IMS. Availability of aggregated raw data in the data storage system is important for manufacturing information for the user at the presentation block. The user can send the query request to the data storage system. The query is processed and returns the response to the query with the requested information. The flow of data that comes from the data storage block to the information presentation block can be defined as outbound data.



**Figure 5.1: Information Manufacturing Process in IMS**

DQ relies on the inbound data level (source to DSS), the data storage level (DSS) and the outbound data level (DSS to information presentation). If, there is a timing mismatch in these three levels, there is a DQ problem in the IMS.

Operational DSS can be used as a source in the IMS. Data of the real world state can manipulate the data of the source DSSs. These source DSSs store the data in the central DSS with a continuous or periodic refreshment frequency method. Waiting of data in the source depend on the usage of the refreshment frequency method in the IMS. Therefore, the time distance between two consecutive refreshment processes can be



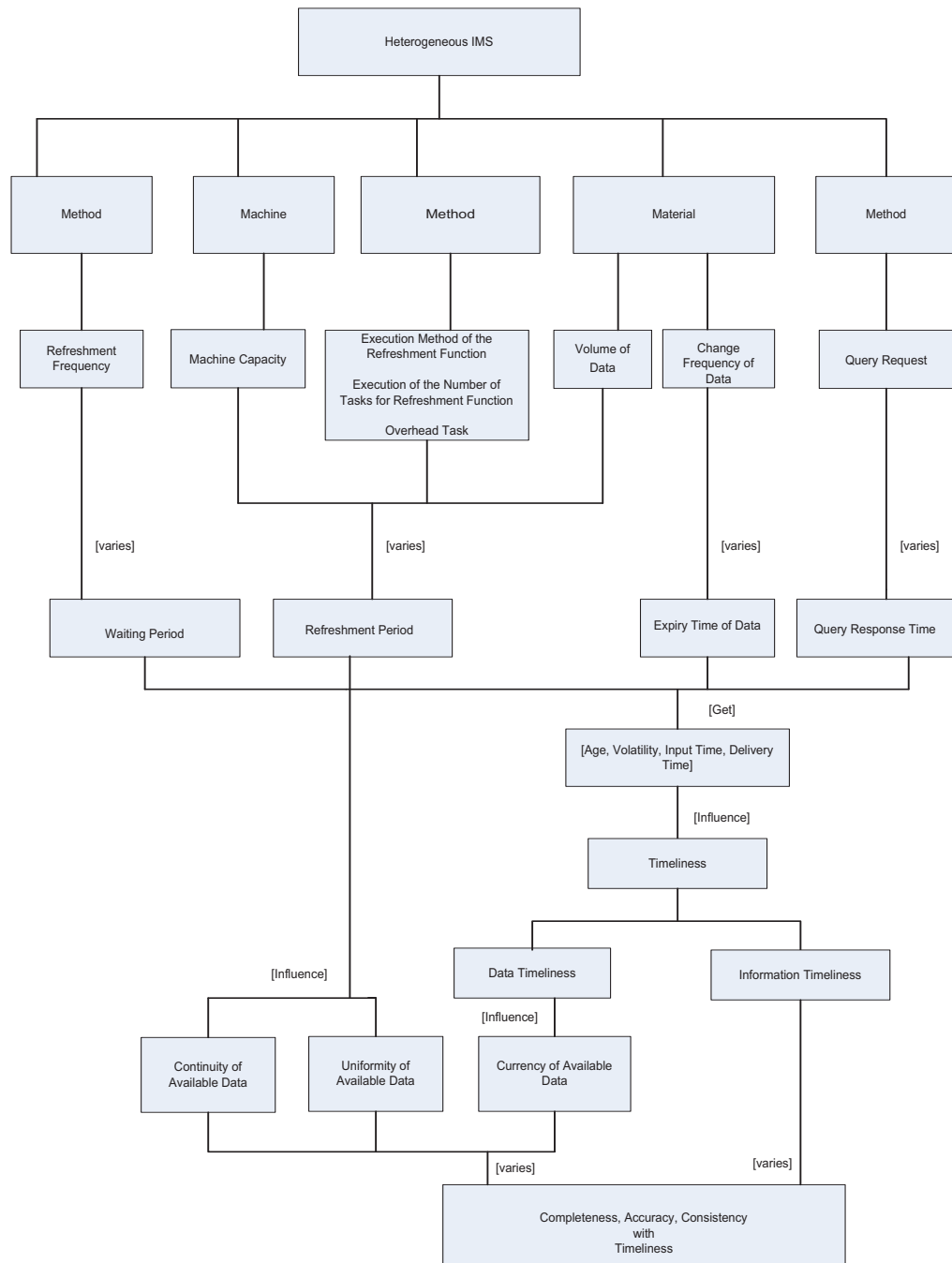
considered as the waiting period of data in the IMS. The waiting period of data varies for the variation of the time distance between two consecutive refreshment processes.

Refreshment period depends on the refreshment processing time of data. The tasks of the refreshment function for making data available was discussed in chapter 2. Refreshment period is regulated by this refreshment function. Data can be available in the IMS by only executing the loading function. Therefore, the execution of the number of tasks for the refreshment function can regulate the refreshment period. Overhead tasks such as transformation, DQ checking etc. can also regulate the refreshment period by including the tasks of the refreshment function. However, if more than one refreshment function is executed in the IMS, then, execution process of the tasks of the refreshment function can control the refreshment period. Further, operational machine capacity and the volume of existing data and newly inserted data can increase or decrease the refreshment time period of the IMS. Therefore, the refreshment time period varies for the machine capacity, volume of data, number of tasks for refreshment function, overhead tasks for refreshment function and the execution method of the tasks of the refreshment function.

Change frequency of the newly inserted or existing data may be long-term or frequently changing. Change frequency of the data is determined by the expiry time. Data of the long expiry time can be recognized as the long term changing data. Expiry time of the frequently changing data will be short. The IMS can contain the mixed (frequently changing + long-term changing) categories of data or the same categories of data. Expiry times of the mixed categories data have not been the same. Similarly, data of the same categories (frequently changing or long-term changing) may not have unique expiry times. Therefore, expiry times of data vary for the change frequency of data.



There is a query response time for each individual query request. Therefore, query response time will be different for each individual query request. So, query response time will also vary.



**Figure 5.2: Changing Process of Data Quality with Timeliness in IMS**



The process of the change of DQ with timeliness is shown in Figure 5.2. In this figure, waiting period, refresh period, query response time and expiry time varies for the factors involved with the heterogeneity of IMS. Waiting period, refresh period, query response time and expiry time influence the DQ constraints such as: continuity of available data; uniformity of available data; data timeliness and information timeliness. Data timeliness influences the currency of available DQ constraint. These DQ constraints in turn affect the DQ dimension such as completeness, accuracy and consistency. Therefore, DQ will vary with timeliness for the variation of the involving factors of heterogeneous IMS.

### **5.2.1 Diversification of Data Quality in Heterogeneous IMS in Theoretical**

#### **Perspective**

Availability of data on time in the DSS of the IMS ensures the quality of time related data. In the DSS of the IMS, if availability is low timeliness is high. On the contrary, if availability is high timeliness is low (Capiello and Helfert, 2008). A timeliness problem could occur for the existence of large volumes of data in a non-indexed DSS (Santos et al., 2008). Availability of data depends on the refresh period. Moreover, the validity of data in the IMS for currency bindings may depend on the refreshment frequency and refreshment period. Continuous refreshment frequency is considered for each type of IMS. Therefore, DQ may vary for the variation of the refresh period. Suppose, that the same data storage software (SQL SERVER, Oracle etc.), the same capacity processor, RAM, Hard Disk or Flash Memory, fixed volume of data and similar change frequency of data has been used for the measurement of the DQ of each type of IMS. And moreover, the time constraint query request is used for each DSS oriented IMS to get



information on time. The heterogeneous IMSs that will be analysed in theoretical aspect are given in Table 5.1.

**Table 5.1: Heterogeneous IMS for Data Quality Analysis in Theoretical Aspect**

<b>DSS Oriented Heterogeneous IMS</b>	<b>Execution Method of Tasks of Refreshment Function</b>	<b>Number of Tasks for Refreshment Function</b>	<b>Execution Method of Refreshment &amp; Query Function</b>	<b>Refreshment Frequency Method</b>	<b>Query Method</b>	<b>Machine Capacity</b>	<b>Volume of Data</b>	<b>Change Frequency of Data</b>
Single DSS Oriented IMS	Sequential	L + Ix	Sequential	Continuous	Time Constraint	High/Low	High/Low	Frequent/ Non-Frequent
Cluster DSS Oriented IMS	Sequential	L + Ix + P	Simultaneous	Continuous	Time Constraint	High/Low	High/Low	Frequent/ Non-Frequent
2-DSS Oriented IMS	Sequential	L, L+Ix	Sequential	Continuous	Time Constraint	High/Low	High/Low	Frequent/ Non-Frequent
3-DSS Oriented IMS	Simultaneous	L + Ix	Simultaneous	Continuous	Time Constraint	High/Low	High/Low	Frequent/ Non-Frequent

A number of variables and terms are used for measuring DQ in a theoretical aspect. The variables and terms that are used for measuring DQ in a theoretical aspect for the heterogeneous IMS are given in Table 5.2.



**Table 5.2: Notations of Variables and Terms for Measuring the Data Quality with Timeliness in Heterogeneous IMS**

Variable and Term	Notation
Age	A
Volatility	V
Loading Period	L
Indexing Period	I <sub>x</sub>
Propagation of Data Period	P
Loading Period (Temporary DSS)	L <sub>tm</sub>
Loading Period (Permanent DSS)	L <sub>p</sub>
Indexing Period (Permanent DSS)	I <sub>xp</sub>
Start of Data Insertion Time from Source	Y
Input Time of Last Data or Data Record of Source	ITL
Input Time of Others Data of Source	IT1, IT2, IT3
Expiry Time	ET
Delivery Time	DT
Completeness, not Completeness	C, ¬C
Accuracy, not Accuracy	Acc, ¬Acc
And	^
Number of times a task execute.	n, n>0
Number of times a data is propagated to the DSS.	m, m>=0

In the IMS, users do not know much about the system, updating time of the data, updating process of the data and so forth. Therefore, the user could send the query request at either the time of the refresh period or after completion of the refresh period. For an example: Suppose, there are a number of data in the source. These data are inserted in DSS with a refreshment process. This process is stopped after the insertion of last data of the source to DSS. Insertion time of data is recognized as the input time



of data. Now, the user sends a query request before the completion of the refreshment process or after the completion of the refreshment process. There is a query response time for the query request. This query response time is known as delivery time. Completeness DQ problem depends on the query response time. If the query is responded to before the insertion time of the last data, the input time of last data will be greater than the delivery time. Therefore, there will be a completeness DQ problem. On the other hand, if the query is responded to after the insertion time of the last data, the input time of the last data will be less than the delivery time. As a result, there will be no completeness DQ problem. Therefore, the level of DQ will depend on whether the query is responded to before or after the completion of the refreshment process.

Accuracy of data relies on whether data is obsolete or not. Obsolescence of data depends on the currency and volatility. The currency of the data can be greater or smaller than the volatility of the data. If the currency of data is smaller than the volatility of data, data will not be obsolete and vice versa. Currency of data varies for the age and the delivery time of the data. On the other hand, volatility of data depends on the expiry time of data. It has been seen in chapter 4 that age is calculated from waiting and refreshment processing period. This is varied in heterogeneous IMS for the variation of refreshment processing and waiting period. Whether data is obsolete or not are assessed after the insertion of data in the DSS of IMS. As a result, accuracy of the data is assessed after the input time of data. Therefore, delivery time after input time and expiry time of data plays the major role in identifying the accurate and inaccurate data. As data does not exist before the insertion of data in the DSS, delivery time before input time is useless for the accuracy assessment of data. Therefore, DQ will rely on



whether the delivery time after input time of the data is greater or smaller than the expiry time of the data.

Firstly: If,  $\text{Input Time} > \text{Delivery Time}$ , all the data of the source are not inserted in the DSS. Therefore, there will be a DQ problem in IMS.

Secondly: If,  $\text{Input Time} < \text{Delivery Time}$ , all data of the source are inserted in the DSS. Therefore, there will not be a DQ problem in IMS.

Thirdly: If,  $\text{Expiry Time} > \text{Delivery Time}$ , timeliness of the inserted data in the DSS is positive. This indicates the non-obsolete-ness of data; therefore, there will not be a DQ problem in IMS.

Fourthly: If,  $\text{Expiry Time} < \text{Delivery Time}$ , timeliness of the inserted data in the DSS is negative. This indicates the obsolete-ness of data; therefore, there will be a DQ problem in IMS.

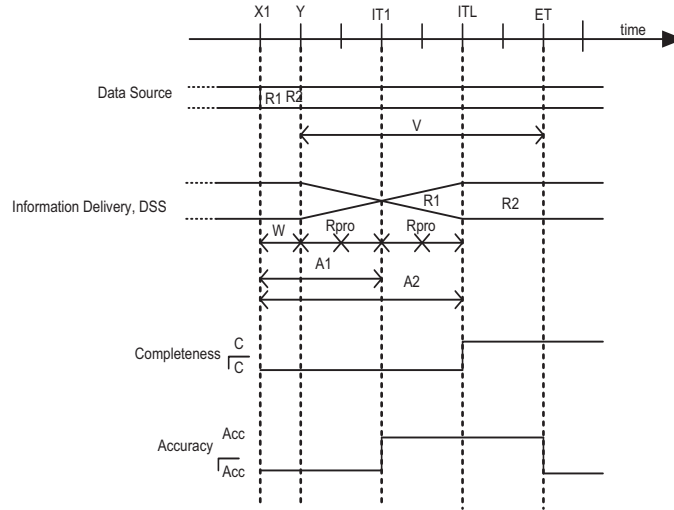
The DQ scenario for the variation of the timeliness factors in IMS is shown in Table 5.3.

**Table 5.3: Data Quality Scenario for Timeliness Factors in IMS**

Input Time VS Delivery Time	Expiry Time VS Delivery Time	Data Quality Dimensions (Completeness, Accuracy)
ITL > DT	ET < DT	$\neg C \wedge \neg \text{Acc}$
	ET > DT	$\neg C \wedge \text{Acc}$
ITL < DT	ET < DT	$C \wedge \neg \text{Acc}$
	ET > DT	$C \wedge \text{Acc}$



The DQ measurement scenario of heterogeneous DSS oriented IMS will be shown illustratively in this chapter. Therefore, Table 5.3 is presented in Figure 5.3.



**Figure 5.3: General DQ Measurement Scenario in IMS**

In the Figure 5.3,

$$IT = \{ IT1, ITL \} \dots \dots \dots (5.1)$$

$$IT = Y + n(Rpro) \dots \dots \dots (5.2)$$

$$A = W + n(Rpro) \dots \dots \dots (5.3)$$

$$C = A + (DT - IT) \dots \dots \dots (5.4)$$

$$V = ET - Y \dots \dots \dots (5.5)$$

Figure 5.3 shows illustratively the DQ measurement scenario of the IMS that provides information from DSS. In this figure, R1 and R2 are two data sets. X1 is the data insertion time in the source. Data insertion is started from source to the DSS at Y for the refreshment process. Therefore, the time between X1 and Y is the waiting time of



data. The data is then refreshed to be available in the DSS. All Data of the source is inserted or input in DSS after completion of the refreshment process of data at ITL. ET is the expiry time of the inserted data. It comes with the source data in the DSS. A1, A2 and V are the age and volatility of data respectively. A1 and A2 are calculated from W and Rpro. Currency is then calculated from age and some other parameters. This is shown in equation 5.4. On the other hand, V is calculated from ET and Y. It is shown in equation 5.5. Information can be delivered at any time from DSS before or after the completion of Rpro.

It is considered that the refreshment frequency is continuous. Therefore, the refreshment process executes continuously until all data of the source are inserted in DSS. It is seen in this figure that when information is delivered at the time of refreshment period before ITL, the delivered information is not complete. The delivered information is complete when information is delivered after ITL. Currency of the data that are inserted at Y is less or more than the volatility of the data. Timeliness of data is calculated from currency and volatility. Therefore, timeliness of data will be positive until the information is delivered after the ET. As a result, the inserted data at ITL or before ITL will be obsolete after ET and hence accurate before ET. Therefore, it can be written that, if,  $ITL < DT < ET$ , then, data is accurate, and if otherwise is inaccurate. Furthermore, it is also seen in this figure that when  $ITL < DT < ET$ , there are no completeness and accuracy DQ problems.

There is a time variation for refreshing period of data in heterogeneous DSS oriented IMS. For example, if only the loading task is considered for the refreshment process, the loading period will be used for calculating the age of the data. For the refreshment process, if only the loading and indexing tasks are considered, age will be calculated by



both the loading and indexing period. Furthermore, if the refreshment period is calculated by the loading, indexing and propagation periods, age will be calculated by adding the time unit of these three periods.

The delivery time of data could be after the end of the refresh period or before the end of refresh period for each type of the IMS and waiting period of data is fixed. Then, if age is calculated from the loading period, the volatility of data could be less or more than the currency calculated by this period. Similarly, if the loading and indexing periods are used for calculating the age, the volatility of data could be less or more than the currency of the data. Volatility of data could be less or more than the currency if age is calculated by the loading, indexing and propagation of the data period.

Now, each individual period of the tasks of the refreshment process is used for the calculation of age, to show the DQ (accuracy) comparison among DSS oriented IMS in a theoretical aspect.

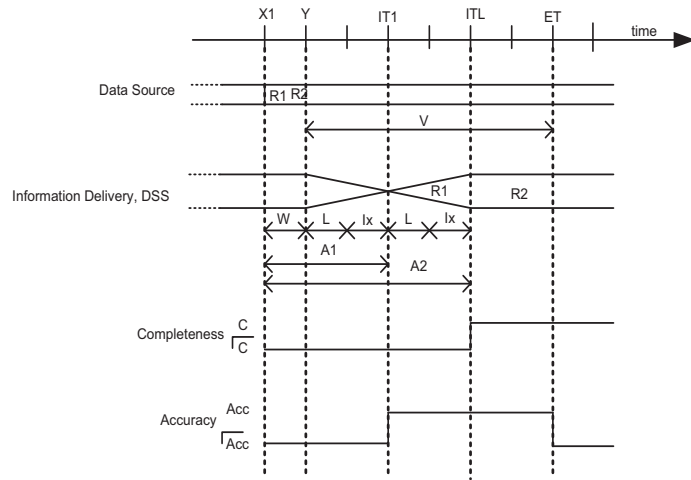
Loading period, indexing period and propagation of data period are the time unit. These periods are not the same. For example: indexing of data needs more time than loading a data. It is because indexing needs to be updated for each piece of data existing in the DSS for loading a new piece of data. This time unit could be millisecond, second, or minute etc.

#### **5.2.1.1 Single DSS Oriented IMS**

Data is manipulated (insert, update) in the single DSS oriented IMS. For availability manipulated data needs to be processed in the DSS. The tasks of the refreshment function of the DSS (loading + indexing) process data to be available for information support. Query requests can be sent before or after the completion of the refreshment



period of a single DSS. Therefore, query can be responded before or after the completion of refreshment period. Information will be delivered on time from the single DSS considering the time constraint query request. However, the single DSS oriented IMS cannot provide continuous required available information for updating data in the DSS. Volatility of data can be greater or smaller than the currency of the data of a single DSS. Furthermore, timeliness of data varies for the variation of the currency of the data. The timeliness value of data determines whether data is current or non-current. Figure 5.4 shows illustratively the DQ measurement scenario of the Single DSS oriented IMS.



**Figure 5.4: DQ Measurement Scenario of Single DSS Oriented IMS**

In the Figure 5.4,

$$IT = \{ IT1, ITL \} \dots \dots \dots (5.6)$$

$$R_{pro} = n(L + Ix) \dots \dots \dots (5.7)$$

$$IT = Y + n(L + Ix) \dots \dots \dots (5.8)$$

$$A = W + n(L + Ix) \dots \dots \dots (5.9)$$



In Figure 5.4,  $X1$  is the data insertion time in the source. Data insertion is started from source to the DSS at  $Y$  for the refreshment process. Therefore, the time between  $X1$  and  $Y$  is the waiting time of data. The data is then refreshed to available in the DSS.  $R_{pro}$  is  $n(L + I_x)$  in this DSS. Data is inserted or input in DSS after the  $R_{pro}$  of data at  $IT1$  and  $ITL$ .  $ET$  is the expiry time of the data. It comes with the source data in the DSS.  $A1$  and  $V$  are the age and volatility of the data respectively.  $A1$  is the age of the  $R1$ .  $A1$  is calculated from  $W$  and  $R_{pro}$ . Similarly,  $A2$  is the age of the  $R2$ .  $A2$  is calculated from the  $W$  and the  $R_{pro}$  of both  $R1$  and  $R2$ . Currency is then calculated from the age. This is shown in equation 5.4. On the other hand,  $V$  is calculated from  $ET$  and  $Y$ . It is shown in equation 5.5. Information can be delivered at any time from DSS before or after the completion of total  $R_{pro}$ .

It is considered that the refreshment frequency is continuous. Therefore, the refreshment process executes continuously until all data of the source are inserted in DSS. It is seen in this figure that when information is delivered before  $ITL$ , the delivered information is not complete. The delivered information is complete when information is delivered after  $ITL$ . Currency of the data that are inserted at  $Y$  is less or more than the volatility of the data. Timeliness of data is calculated from currency and volatility. Therefore, timeliness of data will be positive until the information is delivered after the  $ET$ . As a result, the inserted data at  $ITL$  or before  $ITL$  will be obsolete after  $ET$  and hence accurate before  $ET$ . Therefore, it can be written that, if  $IT1 < DT < ET$ , then, data is accurate, and if otherwise is inaccurate. Further, it is also seen in this figure that when  $ITL < DT < ET$ , there are no completeness and accuracy DQ problems.



### 5.2.1.2 Cluster DSS Oriented IMS

Data can be manipulated in the cluster DSS oriented IMS. The time constraint query request can be sent to the IMS before or after the refreshment processing period to get information on time.

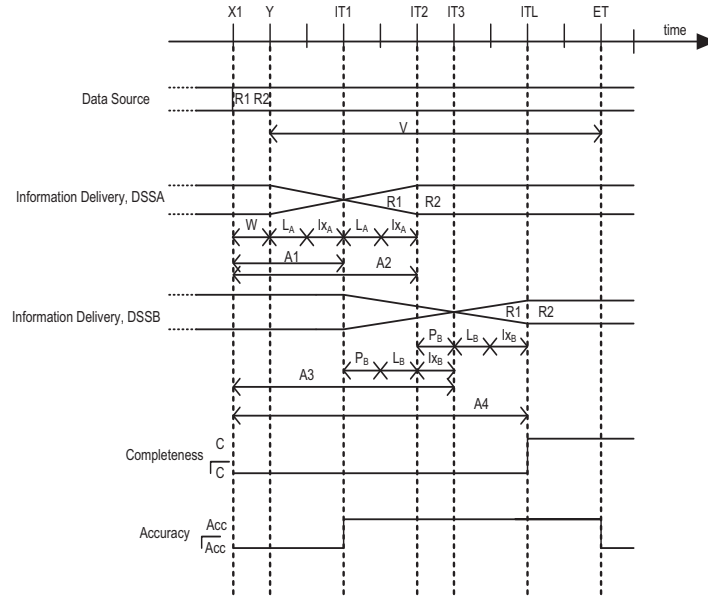
Now, let, R number of DSSs in the cluster DSS oriented IMS. The refreshment processing period of each DSS of the cluster DSS is  $(L + I_x + P)$ . There is no propagation period for the first DSS of the cluster. Therefore,  $P = 0$  time unit for the first DSS of the cluster.

Now, the refreshment processing period for each individual DSS of the cluster DSS oriented IMS can be shown as:

- Refreshment period for 1<sup>st</sup> DSS =  $(L_1 + I_{x_1} + P_1)$  time unit
- Refreshment period for 2<sup>nd</sup> DSS =  $(L_2 + I_{x_2} + P_2)$  time unit
- Refreshment period for 3<sup>rd</sup> DSS =  $(L_3 + I_{x_3} + P_3)$  time unit
- Refreshment period for R<sup>th</sup> DSS =  $(L_r + I_{x_r} + P_r)$  time unit

In the cluster DSS oriented IMS; manipulated data is available in each DSS of the cluster after the completion of the data refreshment processing period. Therefore, when data is inserted in cluster DSS oriented IMS; availability of data for information is not continuous. Volatility of data can be greater or smaller than the currency of the data of a cluster DSS. Furthermore, timeliness of data varies for the variation of the currency of the data. This can make delivered data current or non-current. A DQ measurement scenario of the cluster DSS oriented IMS is shown in Figure 5.5.





**Figure 5.5: DQ Measurement Scenario of Cluster DSS Oriented IMS**

In the Figure 5.5,

$$IT = \{IT1, IT2, IT3, ITL\} \dots \dots \dots (5.10)$$

$$Rpro = n(L + Ix) + mP \dots \dots \dots (5.11)$$

$$IT = Y + (n(L + Ix) + mP) \dots \dots \dots (5.12)$$

$$A = (W + n(L + Ix) + mP) \dots \dots \dots (5.13)$$

Two DSSs are used in Figure 5.5. One is DSSA and another is DSSB. Rpro of DSSA is L and Ix. On the other hand, Rpro of DSSB is L, Ix and P. Therefore, total Rpro of the cluster DSS oriented IMS is  $n(L + Ix) + mP$ . X1 is the data insertion time in the source. Start of data insertion time from source to the DSSA is Y for the refreshment process of the cluster DSS. Therefore, the time between X1 and Y is the waiting time of data. The data is then refreshed to available in the DSSA and DSSB. R1 is inserted or input in DSSA and DSSB after the Rpro of data at IT1 and IT3 respectively. Similarly, R2 is



inserted or input in DSSA and DSSB after the Rpro of data at IT2 and ITL respectively. Therefore, the last data is inserted into DSSB of the cluster at ITL. ET is the expiry time of the data. It comes with the source data in the DSS. V is the volatility of the inserted data at Y. V is calculated from ET and Y. It is shown in equation 5.5. On the other hand, A1 and A2 are the age of the inserted data of DSSA. Similarly, A3 and A4 are the age of the inserted data of DSSB. These are calculated from W and Rpro. Currency is then calculated from the age and some other parameters. This is shown in equation 5.4. Information can be delivered at any time from DSS before or after the completion of total Rpro.

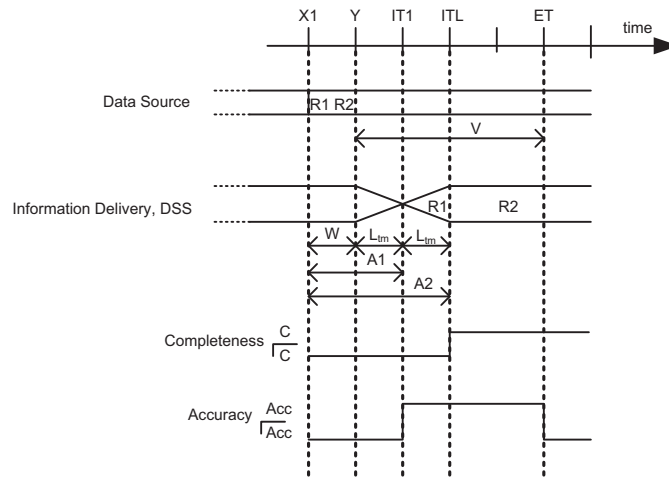
It is considered that the refreshment frequency is continuous. Therefore, the refreshment process executes continuously until all data of the source are inserted in DSS. It is seen in this figure that when information is delivered before ITL, the delivered information is not complete. The delivered information is complete when information is delivered after ITL. Currency of the data that are inserted at Y is less or more than the volatility of the data. Timeliness of data is calculated from currency and volatility. Therefore, timeliness of data will be positive until the information is delivered after the ET. As a result, the inserted data at IT1, IT2, IT3 and ITL will be obsolete after ET and hence accurate before ET. Therefore, it can be written that, if  $IT1 < DT < ET$ , then, data is accurate, and if otherwise is inaccurate. It is also shown in this figure that when  $ITL < DT < ET$ , there are no completeness and accuracy DQ problems.

### **5.2.1.3 2-DSS Oriented IMS**

Manipulated data are processed twice in the 2-DSS. Data are available at DSS (for information) after each refreshment processing. The first refreshment period is shorter



than the second refreshment period. Firstly the refreshment processing is completed in the non-indexed DSS. Therefore, data are available for information after refreshment (loading only) processing period. Conversely, the second refreshment processing is completed in the indexed DSS and, behaves like a single DSS; therefore, (for information), it needs more time to be available. Data waits in the temporary DSS before coming to the indexed DSS. Therefore, the refreshment processing period of data in temporary DSS will be the waiting period of data. The DQ measurement scenario of both non-indexed and indexed DSS of the 2-DSS oriented IMS are shown in Figure 5.6 and Figure 5.7 respectively.



**Figure 5.6: DQ Measurement Scenario of 2-DSS Oriented IMS (Temporary DSS)**

In the Figure 5.6,

$$IT = \{IT1, ITL\} \dots \dots \dots (5.14)$$

$$Rpro = nL_{tm} \dots \dots \dots (5.15)$$

$$IT = Y + nL_{tm} \dots \dots \dots (5.16)$$

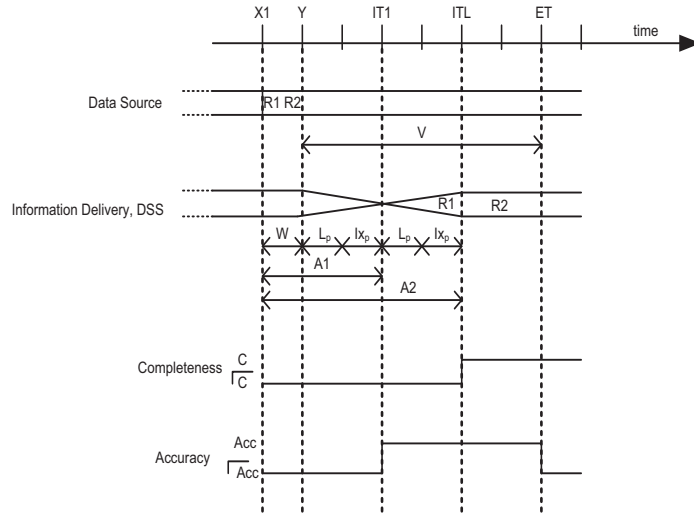
$$A = W + nL_{tm} \dots \dots \dots (5.17)$$



X1 is the data insertion time in the source in the Figure 5.6. Data insertion is started from source to the DSS at Y for the refreshment process. Therefore, the time between X1 and Y is the waiting time of data. The data is then refreshed to available in the DSS. Rpro is  $nL_{tm}$  in this DSS. Data is inserted or input in DSS after the Rpro of data at IT1 and ITL. ET is the expiry time of the data. It comes with the source data in the DSS. A1 and V are the age and volatility of the data respectively. A1 is the age of the R1. A1 is calculated from W and Rpro. Similarly, A2 is the age of the R2. A2 is calculated from the W and the Rpro of both R1 and R2. Currency is then calculated from the age. This is shown in equation 5.4. On the other hand, V is calculated from ET and Y. It is shown in equation 5.5. Information can be delivered at any time from DSS before or after the completion of total Rpro.

It is considered that the refreshment frequency is continuous. Therefore, the refreshment process executes continuously until all data of the source are inserted in DSS. It is seen in this figure that when information is delivered before ITL, the delivered information is not complete. The delivered information is complete when information is delivered after ITL. Currency of the data that are inserted at Y is less or more than the volatility of the data. Timeliness of data is calculated from currency and volatility. Therefore, timeliness of data will be positive until the information is delivered after the ET. As a result, the inserted data at ITL or before ITL will be obsolete after ET and hence accurate before ET. Therefore, it can be written that, if  $IT1 < DT < ET$ , then, data is accurate, and if otherwise is inaccurate. Further, it is also seen in this figure that when  $ITL < DT < ET$ , there are no completeness and accuracy DQ problems.





**Figure 5.7: DQ Measurement Scenario of 2-DSS Oriented IMS (Permanent DSS)**

In the Figure 5.7,

$$IT = \{ IT1, ITL \} \dots \dots \dots (5.18)$$

$$Rpro = n(L_p + Ix_p) \dots \dots \dots (5.19)$$

$$IT = Y + n(L_p + Ix_p) \dots \dots \dots (5.20)$$

$$A = W + n(L_p + Ix_p) \dots \dots \dots (5.21)$$

X1 is the data insertion time in the source in the Figure 5.7. Data insertion is started from source to the DSS at Y for the refreshment process. Therefore, the time between X1 and Y is the waiting time of data. The data is then refreshed to available in the DSS. Rpro is  $n(L_p + Ix_p)$  in this DSS. Data is inserted or input in DSS after the Rpro of data at IT1 and ITL. ET is the expiry time of the data. It comes with the source data in the DSS. A1 and V are the age and volatility of the data respectively. A1 is the age of the R1. Therefore, A1 is calculated from W and Rpro. Similarly, A2 is the age of the R2. Therefore, A2 is calculated from the W and the Rpro of both R1 and R2. Currency is then calculated from the age. This is shown in equation 5.4. On the other hand, V is



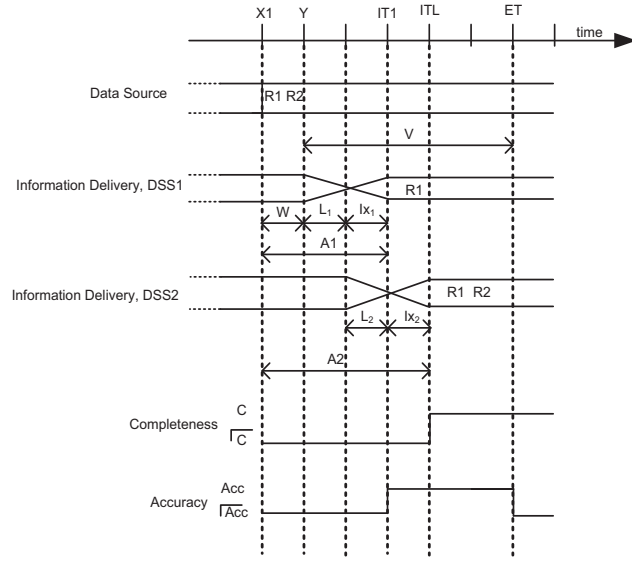
calculated from ET and Y. It is shown in equation 5.5. Information can be delivered at any time from DSS before or after the completion of total Rpro.

It is considered that the refreshment frequency is continuous. Therefore, refreshment process executes continuously until all data of the source are inserted in DSS. It is seen in this figure that when information is delivered before ITL, the delivered information is not complete. The delivered information is complete when information is delivered after ITL. Currency of the data that are inserted at Y is less or more than the volatility of the data. Timeliness of data is calculated from currency and volatility. Therefore, timeliness of data will be positive until the information is delivered after the ET. As a result, the inserted data at ITL or before ITL will be obsolete after ET and hence accurate before ET. Therefore, it can be written that, if  $IT1 < DT < ET$ , then, data is accurate, and if otherwise is inaccurate. Further, it is also seen in this figure that when  $ITL < DT < ET$ , there are no completeness and accuracy DQ problems.

#### **5.2.1.4 3-DSS Oriented IMS**

The tasks of the refreshment function work simultaneously in the 3-DSS. Like the other DSS, the 3-DSS oriented IMS cannot provide continuous available data for the insertion of new data. Furthermore, obsolescence of new data depends on whether the volatility of data is greater or smaller than the refreshment period of data in 3-DSS. Both loading and indexing tasks have to be executed for the availability of data in the 3-DSS. Furthermore, as the time constraint query request is used, information will be delivered on time. DQ scenario of the 3-DSS oriented IMS is shown in Figure 5.8.





**Figure 5.8: DQ Measurement Scenario of 3-DSS Oriented IMS**

In the Figure 5.8,

$$IT = \{ IT1, ITL \} \dots \dots \dots (5.22)$$

$$Rpro = (L + nIx) \dots \dots \dots (5.23)$$

$$IT = Y + (L + nIx) \dots \dots \dots (5.24)$$

$$A = W + (L + nIx) \dots \dots \dots (5.25)$$

X1 is the data insertion time in the source in the Figure 5.8. Data insertion is started from source to the DSS at Y for the refreshment process. Therefore, the time between X1 and Y is the waiting time of data. The data is then refreshed to available in the DSS. Rpro is  $(L + nIx)$  in this DSS. Data is inserted or input in DSS after the Rpro of data at IT1 and ITL. ET is the expiry time of the data. It comes with the source data in the DSS. A1 and V are the age and volatility of the data respectively. A1 is the age of the R1. A1 is calculated from W and Rpro in DSS1. Similarly, A2 is the age of the R1 and R2 in DSS2. A2 is calculated from the W and the combined Rpro of DSS1 and DSS2.



Currency is then calculated from the age. This is shown in equation 5.4. On the other hand,  $V$  is calculated from  $ET$  and  $Y$ . It is shown in equation 5.5. Information can be delivered at any time from DSS before or after the completion of total Rpro.

It is considered that the refreshment frequency is continuous. Therefore, the refreshment process executes continuously until all data of the source are inserted in DSS. It is seen in this figure that when information is delivered before ITL, the delivered information is not complete. The delivered information is complete when information is delivered after ITL. Currency of the data that are inserted at  $Y$  is less or more than the volatility of the data. Timeliness of data is calculated from currency and volatility. Therefore, timeliness of data will be positive until the information is delivered after the  $ET$ . As a result, the inserted data at ITL or before ITL will be obsolete after  $ET$  and hence accurate before  $ET$ . Therefore, it can be written that, if  $IT1 < DT < ET$ , then, data is accurate, and if otherwise is inaccurate. Further, it is also seen in this figure that when  $ITL < DT < ET$ , there are no completeness and accuracy DQ problems.

### **5.2.2 Measurement of Data Quality in Heterogeneous Simulated Information Manufacturing System (IMS)**

The real world IMS is simulated for measuring the change of DQ with timeliness. In the real world, various kinds of sources are available for generating or manipulating data in the IMS. Data could be generated automatically from the sources. Call detail record generation in the telecommunication industry is an example of auto generation of data from the sources. Data may be manipulated (insert, delete, update) manually from the sources. Manipulation of stock exchange data from the brokerage house workstation or manipulation of banking data from the branch of the bank are



examples of manual manipulation of data sources. Operational database (ODS) of an organization can act as the source of the IMS. Integration of sources can be designed in different degree level in the IMS. As seen in chapter 3, the degree of integration varies in the cluster DSS oriented IMS for the master-slave DSS and no master-slave DSS. However, the execution of the number of tasks for the refreshment function for each type of DSS (master-slave DSS or no master-slave DSS) is the same. Therefore, higher degree integration of sources is considered for the simulated IMS. The execution methods of the refreshment and the query function are considered for the DSS of the simulated IMS. The execution method of the tasks of the refreshment function and the number of tasks for the execution of refreshment task and the overhead tasks are also considered in the simulated IMS. Machine capacity, volume of data and the change frequency of data are also included in the real world IMS. These are also considered in the simulated IMS. Non-time constraint query method is applied in the experiment of simulated IMS. Therefore, the simulated IMSs considered for the experiment are given in Table 5.4. These simulated IMS will show the change of DQ with timeliness in the IMS.

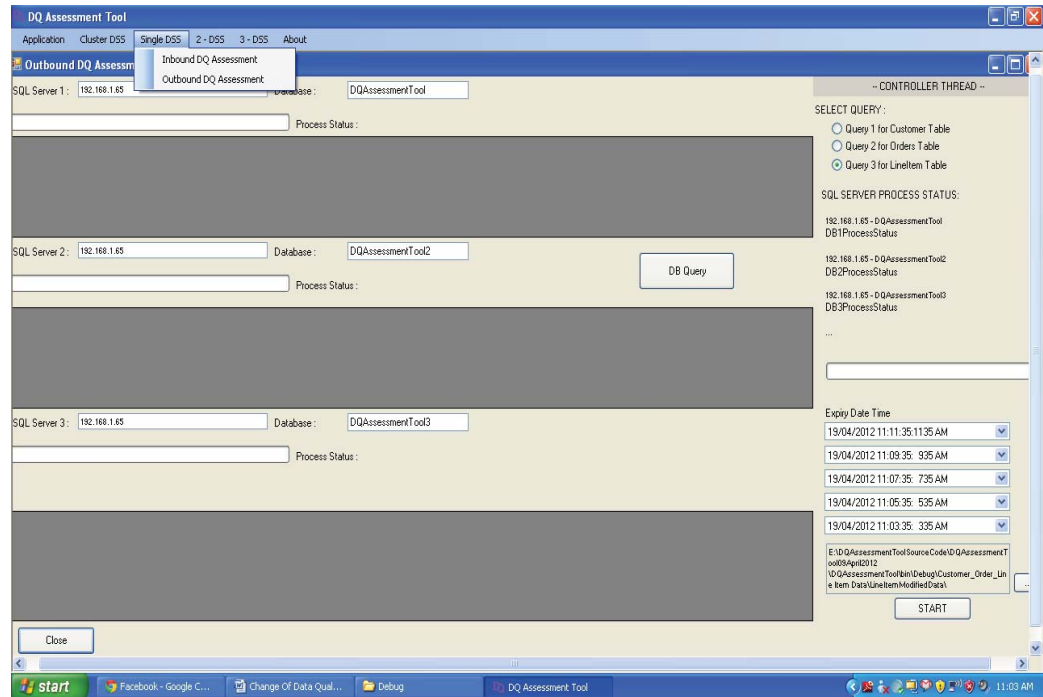


**Table 5.4: Heterogeneous IMS for the Data Quality Measurement in Experimental Aspect**

DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Execution Method of Refreshment & Query Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data
Single DSS Oriented Heterogeneous IMS	Sequential	$L + Ix$	Sequential	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent
	Sequential	$L + Ix$	Sequential	No DQ Checking	Continuous	Non-Time Constraint	Low	High	Non-Frequent
	Sequential	$L + Ix$	Sequential	No DQ Checking	Non-Continuous	Non-Time Constraint	High	High	Non-Frequent
	Sequential	$L + Ix$	Sequential	DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent
Cluster DSS Oriented IMS	Sequential	$L + Ix + P$	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent
2-DSS Oriented IMS	.....	L	Sequential	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent
3-DSS Oriented Heterogeneous IMS	Simultaneous	$L + Ix$	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent
	Simultaneous	$L + Ix$	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	High	Frequent
	Simultaneous	$L + Ix$	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	Low	Non-Frequent

Now, DQ assessment tool and setup of the experiments will be discussed. The frontend of the tool holds the selector of the DSS based simulated IMS. Any of the DSS based simulated IMS can be selected for making the experiment of particular simulated IMS. For an example: if we want to make experiments for the 3-DSS based simulated IMS, source data button and query button have to be pressed for manipulation (insertion) of data from the source and the presenting of the required information respectively. Others DSS based simulated IMS have the same source data button and a query button for the manipulation of data from the source and presentation of information for the query function respectively.





**Figure 5.9: Data Quality Assessment Tool for IMS**

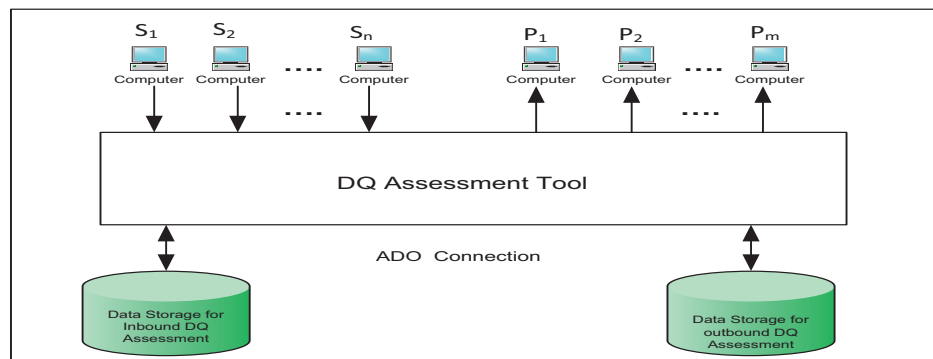
The database contains inbound source data to the data storage system of the simulated IMS. SQL server database software is used for containing the manipulated inbound data. Database access is realized using ADO technology. ADO maintains the connectivity between the frontend and the database backend.

There is a business end between the frontend and the database backend. Each task for execution in the frontend and the database backend are written in the business end. Inbound and outbound DQ assessment function for measuring the DQ of each DSS based simulated IMS works in the business end. This assessment function is written by following the assessment function presented in chapter 4. Timeliness calculation of each DSS based simulated IMS is also written to the business end by following the timeliness function also explained in chapter 4. Refreshment frequency controlling method is also written in the business end. Replication method for cluster DSS oriented simulated IMS is written to this business end. Furthermore, the algorithms of the 3-DSS



oriented simulated IMS described in chapter 3 are implemented in the business end for the execution of the 3-DSS oriented simulated IMS.

There were N numbers of sources in the source block to manipulate data in the data storage system. M numbers of presentation interface were used for the presentation block to retrieve data from the data storage system. The experimental setup of the DQ assessment is shown in Figure 5.10.



**Figure 5.10: Experimental Setup of Data Quality Assessment of Heterogeneous IMS**

A local area network is needed for the DQ assessment of cluster DSS oriented IMS. In this cluster DSS oriented IMS; each data storage system will be interconnected with each other in the network.

Measurement of DQ for both inbound and outbound data helps to show the change of DQ with timeliness in the simulated IMS. Mixed quality data (good + poor) are stored in the individual sources. A benchmark database is created by collecting data from TPC-H where data are 100% complete and 100% accurate. Both inbound and outbound data are compared with the data of the benchmark database for measuring the DQ. Data is inserted sequentially from the sources to the DSS for each experiment of the simulated IMS. For example: when the data insertion from source 1 finishes the data



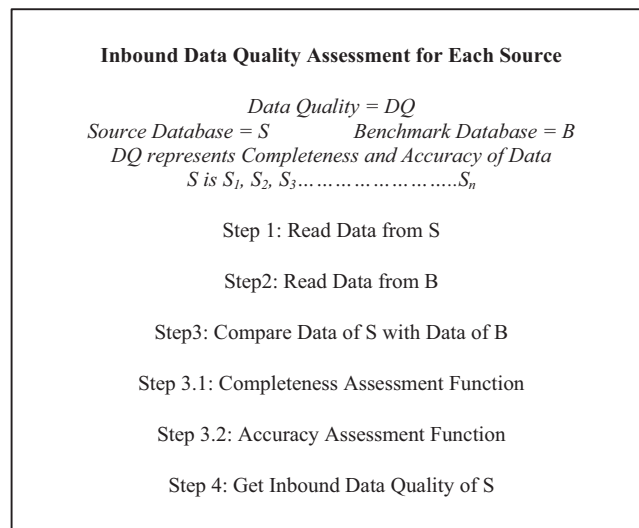
insertion from source 2 begins and so on. A Query is executed in the DSS to get outbound data from the simulated IMS. The query is executed in the TPC-H LINEITEM Table. The query that is executed in the experiment of each of the IMS for outbound data is as follows:

```

SELECT
L_ORDERKEY,L_LINENUMBER,L_QUANTITY,L_EXTENDEDPRICE,L_DISCO
UNT,L_LINESTATUS,L_SHIPDATE,L_COMMITDATE,L_RECEIPTDATE,L_SHI
PINSTRUCT,L_SHIPMODE,L_COMMENT
FROM LINEITEM

```

Inbound DQ assessment for measuring inbound DQ is completed before storing the data in the DSS of the simulated IMS for outbound data. The algorithm in Figure 5.11 shows the inbound DQ assessment procedure for getting the inbound DQ result from the simulated IMS.



**Figure 5.11: Inbound Data Quality Assessment Algorithm**

The inbound DQ assessment algorithm of Figure 5.11 is now described. In this algorithm, the data quality, source and the benchmark database are indicated by DQ, S



and B respectively. DQ represents completeness and accuracy DQ dimensions. Multiple source data will be stored in the DSS of the IMS. The multiple sources are represented as  $S_1, S_2, S_3, \dots, S_n$ . The benchmark database is for inbound DQ assessment. In step 1, data of S will be read. S could be single or multiple sources. In step 2, benchmark data are read from the benchmark database. In step 3, the source data will be compared with the benchmark database data by using the completeness and accuracy DQ assessment function for measuring inbound DQ. In step 4, the percentage of complete and accurate data in the source or sources can be computed. Completeness and accuracy percentage of inbound data in the source or sources are given in Table 5.5.

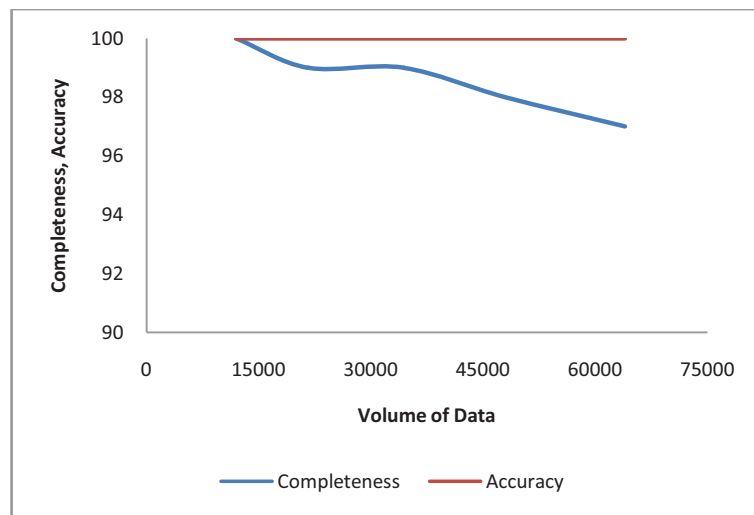
**Table 5.5: Inbound Data Quality Assessment**

Source	Volume of Data	Completeness (%)	Accuracy (%)
$S_1$	12000	100	100
$S_1 + S_2$	21600	99	100
$S_1 + S_2 + S_3$	34404	99	100
$S_1 + S_2 + S_3 + S_4$	48000	98	100
$S_1 + S_2 + S_3 + S_4 + S_5$	63984	97	100

There should be a total 79980 data in five sources. The query is not executed to get all data of the DSS as outbound data. Therefore, all data of source or sources are not assessed for measuring the DQ of inbound data. The source data are compared with the benchmark database. There should be 12000 data in the source 1 and, in the assessment, 12000 data are found. In addition, there was no inaccurate data in source 1. As a result, both completeness and accuracy of inbound data were 100%. On the other



hand, five data sources are assessed. There should be 63984 data in five sources. This volume of data is not found in five sources. Furthermore, there was no inaccurate data to the five data sources. As a result, completeness and accuracy of data are 97% and 100% respectively in the five sources. There should be 21600, 34404 and 48000 in 2 sources, 3 sources and 4 sources respectively. This volume of data is also not found in these sources. Moreover, no inaccurate data was in these sources. As a result, completeness and accuracy of the 2 sources, 3 sources and 4 sources data are (99%, 100%), (99%, 100%) and (98%, 100%) respectively. DQ of the inbound data of the different volumes is shown in Figure 5.12.



**Figure 5.12: Data Quality Assessment Graph for Inbound Data**

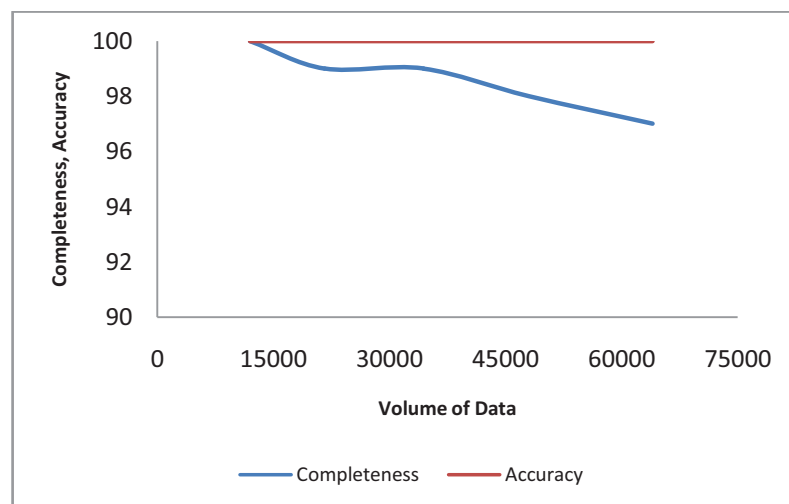
Furthermore, if there is no time related DQ problem in the IMS, the outbound DQ problem = inbound DQ problem. Therefore, an experiment is done where the DQ problem for time related factors are not considered. Data is inserted from the sources one after another and the query given above is executed 5 times. Each query is executed after the completion of data insertion from each of the individual source. Therefore, data are inserted from source 1 and the query is executed to get the outbound data.



These outbound data are then assessed. The same DQ result of the inbound data is found for the outbound data. Hence, data of source 2 is inserted. Now, source 1 and source 2 data is in the data storage system. Now, the quality of the outbound data is again assessed. The quality result of outbound data is the same as the quality result of the inbound data of source 1 plus source 2. The rest of the source data are stored in the same process and got the outbound DQ result as the same as the inbound DQ result. These results are shown in Table 5.6 and presented as a graph in Figure 5.13.

**Table 5.6: Outbound Data Quality Assessment**

Query	Volume of Data	Completeness (%)	Accuracy (%)
1	12000	100	100
2	21600	99	100
3	34404	99	100
4	48000	98	100
5	63984	97	100

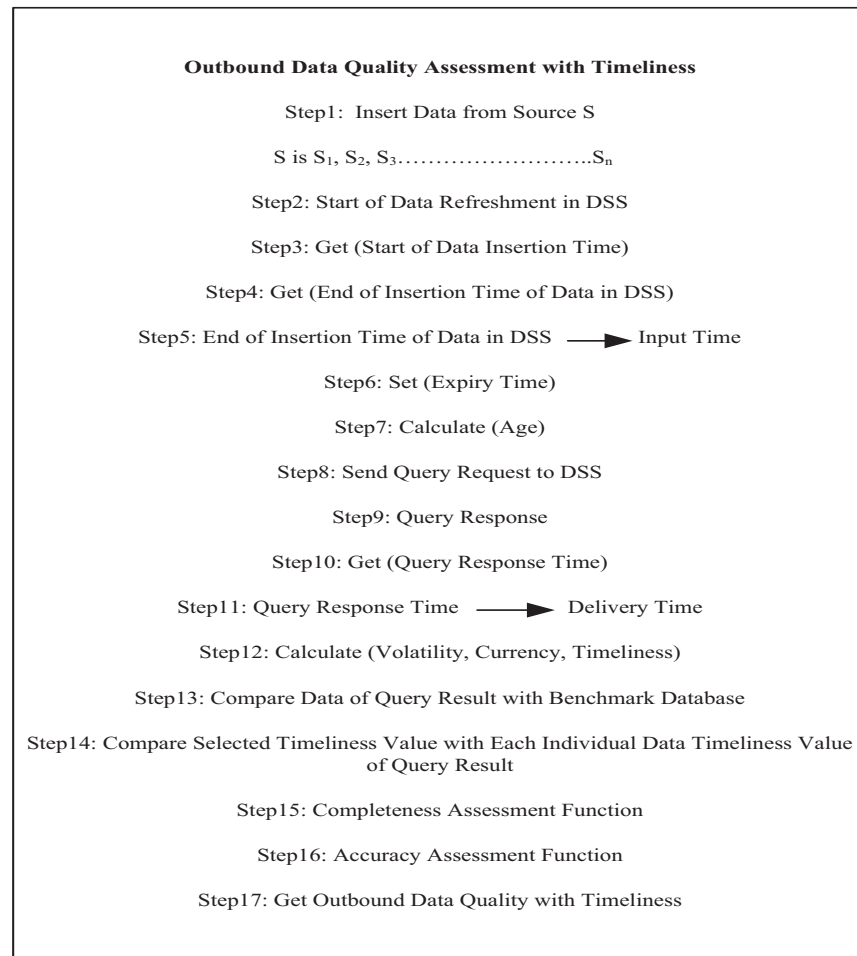


**Figure 5.13: Data Quality Assessment Graph for Outbound Data**



Outbound DQ assessment procedure for each simulated IMS will be described. Refreshment process will be started for inserting data in the DSS. Each inserted data set from sources will get an insertion time for outbound data or query result assessment. End of the refreshment of a data set will be the input time of the data of that set. Expiry time for each inserted data will also be set. When the query request is sent, there will be a query response time. This query response time will be counted as delivery time. Age, volatility, currency and timeliness will be calculated from among the time value of data. Subsequently, each individual timeliness value is compared with the user defined timeliness value to get obsolete data. The timeliness value for each experiment is set at .3. Each query result of the user is compared with the benchmark database. Therefore, getting the outbound DQ (completeness, accuracy) with timeliness, from the accuracy and completeness DQ assessment function. Algorithm for this outbound DQ assessment with timeliness is given in Figure 5.14.





**Figure 5.14: Outbound Data Quality Assessment Algorithm**

### 5.2.2.1 Single DSS Oriented Simulated Information Manufacturing System (IMS1)

This IMS is designed for storing the source data in one single DSS. The source of the simulated IMS is in database files. The tasks of the refreshment function execute sequentially in this simulated IMS1. Furthermore, the query function executes to show information in the presentation block. The query and refreshment function executes sequentially in the IMS1. Four experiments are done in the single DSS oriented heterogeneous IMS. Experiment number of the single DSS oriented heterogeneous IMS is given in Table 5.7.



**Table 5.7: Experiment Number of Single DSS Oriented Heterogeneous IMS**

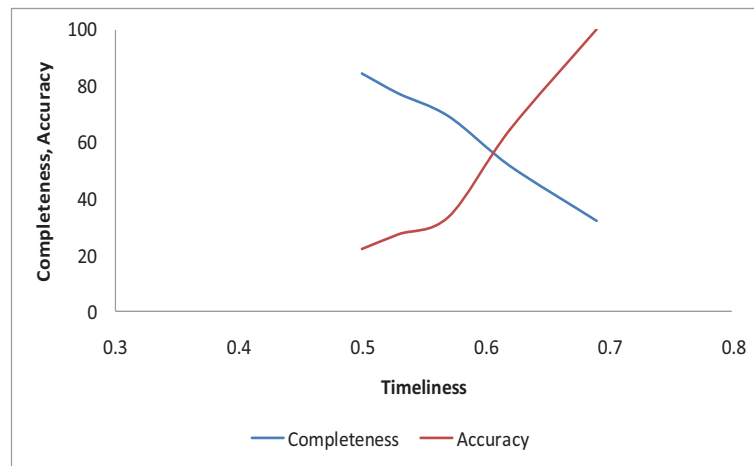
Single DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Execution Method of Refreshment & Query Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Experiment Number
IMS1.1	Sequential	L + Ix	Sequential	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	1
IMS1.2	Sequential	L + Ix	Sequential	No DQ Checking	Continuous	Non-Time Constraint	Low	High	Non-Frequent	2
IMS1.3	Sequential	L + Ix	Sequential	No DQ Checking	Non-Continuous	Non-Time Constraint	High	High	Non-Frequent	3
IMS1.4	Sequential	L + Ix	Sequential	DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	4

Experiment 1 is done for IMS1.1. In the IMS1.1 non-time constraint query method is used. Machine capacity, volume of data and the change frequency of data is high. High volume of data (79980) is used in this experiment. Change frequency of data is non-frequent which means that the volatility or validity of data is high. No overhead task such as, DQ checking is conducted in this IMS. The refreshment frequency method is continuous. The number of tasks for refreshment function is loading and indexing and the execution method of the tasks for refreshment function and the execution method of refreshment and query functions are sequential. The experimental result for this IMS is given in Table 5.8.



**Table 5.8: Outbound Data Quality with Timeliness for IMS1.1**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2488240	10:30:42	10:35:00	258983	836063	.69	32.20	100.00
2	0-3015280	10:30:42	10:35:58	316730	836063	.62	51.90	64.56
3	0-3347490	10:30:42	10:36:40	358966	836063	.57	69.30	33.87
4	0-3607960	10:30:42	10:37:14	392136	836063	.53	77.27	27.57
5	0-4012690	10:30:42	10:37:39	417890	836063	.50	84.30	22.50



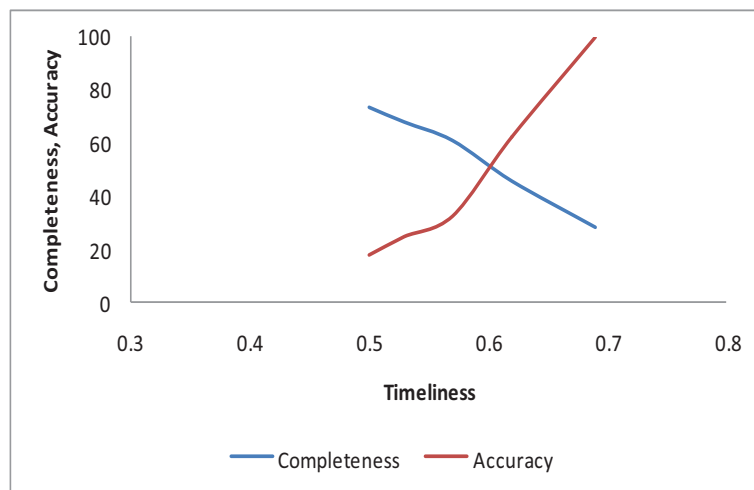
**Figure 5.15: Outbound Data Quality with Timeliness Graph for IMS1.1**

There is no difference between IMS1.1 and IMS1.2 except the capacity of the machine. In the IMS1.2, a low capacity machine is used. Therefore, experiment 2 will be used for measuring the DQ of IMS1.2. The experimental result of IMS1.2 is given in Table 5.9.



**Table 5.9: Outbound Data Quality with Timeliness for IMS1.2**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2522240	12:08:04	12:12:22	258630	836063	.69	28.37	100.00
2	0-3145600	12:08:04	12:13:20	316590	836063	.62	45.80	62.45
3	0-3447490	12:08:04	12:14:03	359290	836063	.57	60.78	32.62
4	0-3728960	12:08:04	12:14:36	392580	836063	.53	67.58	24.56
5	0-4121280	12:08:04	12:15:00	416287	836063	.50	73.40	18.03

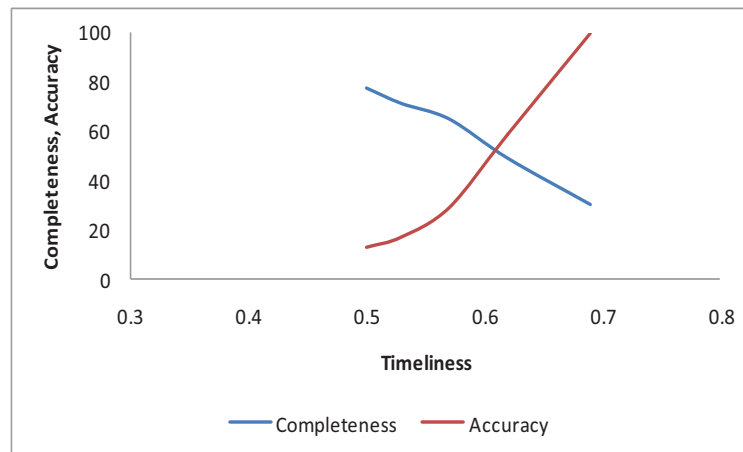
**Figure 5.16: Outbound Data Quality with Timeliness Graph for IMS1.2**

The non-continuous refreshment frequency method is used in the IMS1.3. Some data are inserted in the IMS with a refreshment process. Then, the refreshment process is stopped. After a certain period, refreshment process is restarted. The other parameters of the factors of the heterogeneous IMS are the same as the IMS1.1. This is experiment number 3. The experimental result of this IMS is given in Table 5.10.



**Table 5.10: Outbound Data Quality with Timeliness for IMS1.3**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2774280	16:15:07	16:19:24	257737	836063	.69	30.40	100.00
2	0-3303540	16:15:07	16:20:23	316290	836063	.62	48.82	58.62
3	0-3635620	16:15:07	16:21:04	357828	836063	.57	64.72	28.76
4	0-3932870	16:15:07	16:21:39	392859	836063	.53	71.20	16.52
5	0-4342630	16:15:07	16:22:03	416382	836063	.50	77.60	12.40

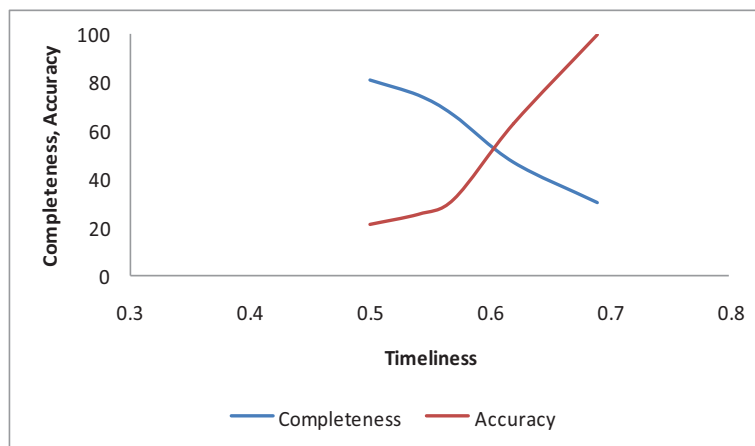
**Figure 5.17: Outbound Data Quality with Timeliness Graph for IMS1.3**

DQ checking for inbound data is done in experiment 4. This experiment is for the IMS 1.4. In this experiment inbound DQ is compared with the benchmark database before inserting data in the DSS. The purpose of this DQ checking is to get the DQ with timeliness after adding overhead task with the tasks of the refreshment function. All other parameters of the factors of the heterogeneous IMS are the same as in the IMS 1.1. The experimental result of the IMS1.4 is given in Table 5.11.



**Table 5.11: Outbound Data Quality with Timeliness for IMS1.4**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2527680	11:05:12	11:09:19	257123	836063	.69	30.38	100.00
2	0-3125300	11:05:12	11:10:29	317028	836063	.62	47.05	62.76
3	0-3478490	11:05:12	11:11:09	357980	836063	.57	66.40	31.67
4	0-3767960	11:05:12	11:11:44	392568	836063	.53	74.60	25.57
5	0-4143800	11:05:12	11:12:10	418230	836063	.50	81.36	20.78



**Figure 5.18: Outbound Data Quality with Timeliness Graph for IMS1.4**

#### **5.2.2.2 Cluster DSS Oriented Simulated Information Manufacturing System (IMS2)**

Machine capacity, change frequency of data, execution method of the refreshment function and volume of data for the insertion of IMS2 is the same as in the experiment of IMS1.1. No overhead task such as DQ checking is included in this experiment. Further, continuous refreshment frequency method and non-time constraint query method are also used in the experiment of IMS2. Number of tasks for the refreshment function and the execution method of the refreshment and query function vary in IMS



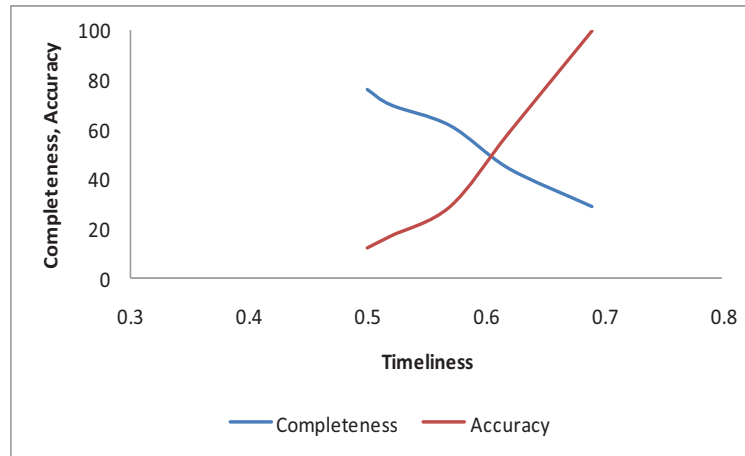
2. The source data are stored in the master DSS. These data are then propagated from master DSS to the four slave DSS. This is done using the replication technique. The immediate propagation method is used for propagating data immediately. The query function executes simultaneously with the refreshment process of the five workstations.

Table 5.12 represents the DQ result of experiment 1 for the IMS2. There was no pre-stored data in any of the DSS of the IMS2 in this experiment. Data is first stored in the master DSS of the IMS2. This data are then propagated to the first, second, third and fourth slave DSS of the IMS2. Therefore, five users sent the query request to four slave DSS at five different times for measuring completeness and accuracy with timeliness in the IMS2.

**Table 5.12: Outbound Data Quality with Timeliness for IMS2**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	5-256458 5	20:12:01	20:16:20	259135	836063	.69	28.38	100.00
2	5-316628 5	20:12:01	20:17:18	317626	836063	.62	44.27	58.75
3	5-354786 5	20:12:01	20:17:59	358966	836063	.57	61.52	28.80
4	5-384856 5	20:12:01	20:18:40	399960	836063	.53	69.76	16.87
5	5-416426 5	20:12:01	20:18:59	418366	836063	.50	76.35	12.05





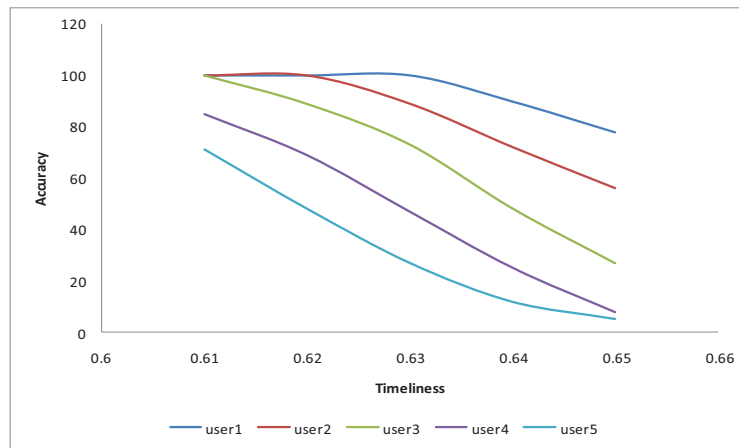
**Figure 5.19: Outbound Data Quality with Timeliness Graph for IMS2**

Another experiment is conducted in the IMS2 for showing the consistency DQ problem with timeliness. In this experiment, there was pre-stored data in each DSS of the cluster of the IMS2. These pre-stored data were to update to do this experiment. This pre-stored data were used to first delete and then new data were inserted at the same time in the master DSS of the IMS2. Five users sent the query request in the DSS of five computers in the cluster DSS of the IMS2. The following result was obtained from this experiment.

**Table 5.13: Outbound Data Quality with Timeliness for IMS2 (Consistency DQ Problem)**

Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	User1		User2		User3		User4		User5	
						C	Acc	C	Acc	C	Acc	C	Acc	C	Acc
0-71520 0	19:10:07	19:12:46	159127	458760	.65	99	78	100	56	100	27	100	8	100	5
0-75830 0	19:10:07	19:12:50	163780	458760	.64	100	90	100	72	100	48	100	25	100	12
0-78560 0	19:10:07	19:12:53	166789	458760	.63	100	100	100	89	100	73	100	47	100	27
0-85786 0	19:10:07	19:12:58	171540	458760	.62	100	100	100	100	100	89	100	69	100	48
0-92130 0	19:10:07	19:13:04	177450	458760	.61	100	100	100	100	100	100	100	85	100	71





**Figure 5.20: Outbound Data Quality with Timeliness Graph for IMS2 (Consistency DQ Problem)**

It can be seen with timeliness, that completeness of the data was almost 100% of the query request for each user. Accuracy however, is varied though each user sent the query request at the same time. It is noticeable that there were small percentages of accurate data of user5 when timeliness was .65. Percentage of accuracy of the other four users is also different when timeliness is .65. Accuracy also varied for all other timeliness value (.64, .63, .62, .61). At timeliness .61, three users got the same query results. This shows that users of the IMS2 did not get a similar query result. Users getting different query results, indicates that there is a consistency DQ problem in the IMS2.

#### **5.2.2.3 2-DSS Oriented Simulated Information Manufacturing System (IMS3)**

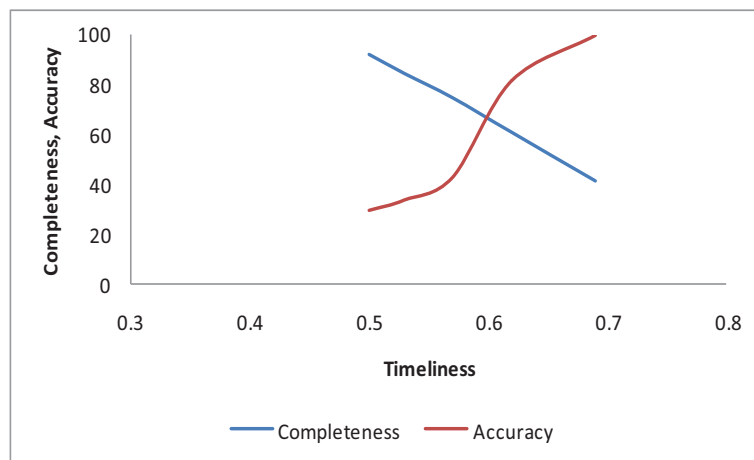
Refreshment frequency method, execution method of the refreshment and query function and the query method of IMS3 are the same as the simulated IMS1.1. Further, machine capacity, change frequency of data and volume of data for insertion are also the same as the IMS1.1. No overhead task such as DQ checking is included in this experiment. Two DSSs are created in the simulated IMS3. One DSS has been created where only inbound data will be loaded and no indexing of data will be done to make



data available in the simulated IMS3. Another DSS will execute both the loading and indexing tasks for making inbound source data available in the simulated IMS3. The experimental result of the temporary table of IMS3 is shown in Table 5.14.

**Table 5.14: Outbound Data Quality with Timeliness for IMS3**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2283040	21:40:00	21:44:18	258998	836063	.69	41.60	100.00
2	0-2755820	21:40:00	21:45:17	317130	836063	.62	60.78	81.56
3	0-3159760	21:40:00	21:45:58	358905	836063	.57	75.08	42.82
4	0-3378400	21:40:00	21:46:33	393127	836063	.53	84.82	33.56
5	0-3655870	21:40:00	21:46:57	417980	836063	.50	92.32	29.76



**Figure 5.21: Outbound Data Quality with Timeliness Graph for IMS3**

In this experiment, the percentage of completeness and accuracy with timeliness of the temporary DSS of the IMS3 is shown. Completeness and accuracy results with timeliness of permanent DSS of the IMS3 will be the same as IMS1.



#### 5.2.2.4 3-DSS Oriented Simulated Information Manufacturing System (IMS4)

This IMS is designed for storing the source data in the three DSSs. The database files that are used as sources in the IMS1 will be used for the IMS4. The tasks of the refreshment function execute simultaneously in the simulated IMS4. Further, execution method of the refreshment and the query function is simultaneous in this simulated IMS. Three experiments are conducted in the 3-DSS oriented IMS. Experiment number of the 3-DSS oriented heterogeneous IMS is given in Table 5.15.

**Table 5.15: Experiment Number of 3-DSS Oriented Heterogeneous IMS**

3-DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Execution Method of Refreshment & Query Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Experiment Number
IMS4.1	Simultaneous	L + Ix	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	1
IMS4.2	Simultaneous	L + Ix	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	High	Frequent	2
IMS4.3	Simultaneous	L + Ix	Simultaneous	No DQ Checking	Continuous	Non-Time Constraint	High	Low	Non-Frequent	3

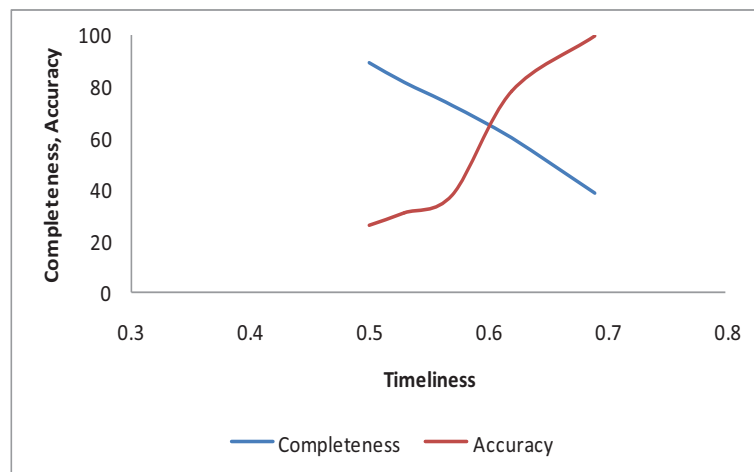
Experiment 1 is the DQ experiment for IMS4.1. In this IMS the non-time constraint query method and continuous refreshment frequency method are used. Machine capacity, volume of data and the change frequency of data is high. Like the other IMS, high volume of data (79980) is used in this experiment. Change frequency of data is non-frequent in this IMS. No overhead task such as, DQ checking is done in this IMS. In the IMS4.1, the execution method of the refreshment and query function and the executing method of the tasks of the refreshment function will be simultaneous. Furthermore, the number of tasks for refreshment function is (loading and indexing) the



same as the number of tasks work in the refreshment function of the heterogeneous IMS of IMS1. The experimental result for the IMS4.1 is given in Table 5.16.

**Table 5.16: Outbound Data Quality with Timeliness for IMS4.1**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2369360	10:59:14	11:03:33	259120	836063	.69	38.82	100.00
2	0-2785800	10:59:14	11:04:31	317730	836063	.62	59.80	78.08
3	0-3248270	10:59:14	11:05:13	359507	836063	.57	72.96	37.90
4	0-3457960	10:59:14	11:05:46	392949	836063	.53	81.73	30.80
5	0-3722630	10:59:14	11:06:12	418003	836063	.50	89.40	26.30



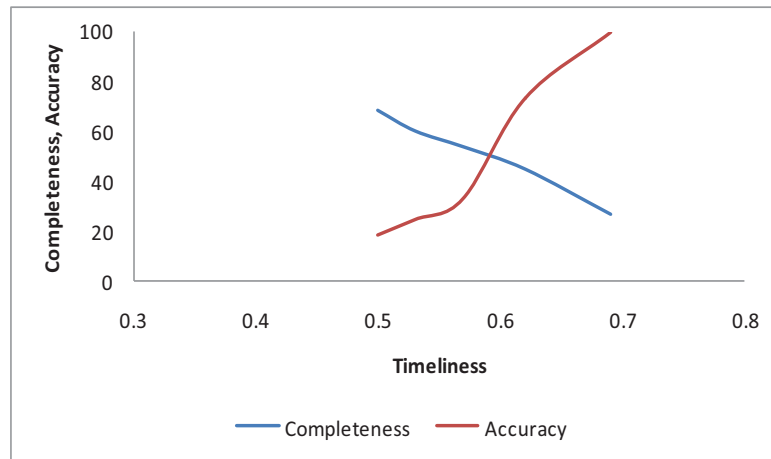
**Figure 5.22: Outbound Data Quality with Timeliness Graph for IMS4.1**

The frequent change frequency of data or the low volatility value of data is used in the IMS 4.2. The other parameters of the factors of the heterogeneous IMS are the same as the IMS4.1. This is experiment number 3. The experimental result of this IMS is given in Table 5.17.



**Table 5.17: Outbound Data Quality with Timeliness for IMS4.2**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-1824280	09:15:07	09:18:59	232737	750765	.69	26.80	100.00
2	0-2563540	09:15:07	09:19:52	285290	750765	.62	44.82	73.62
3	0-2901570	09:15:07	09:20:29	322828	750765	.57	53.72	32.76
4	0-3312870	09:15:07	09:20:59	352859	750765	.53	60.20	24.52
5	0-3562630	09:15:07	09:21:22	375382	750765	.50	68.40	18.30

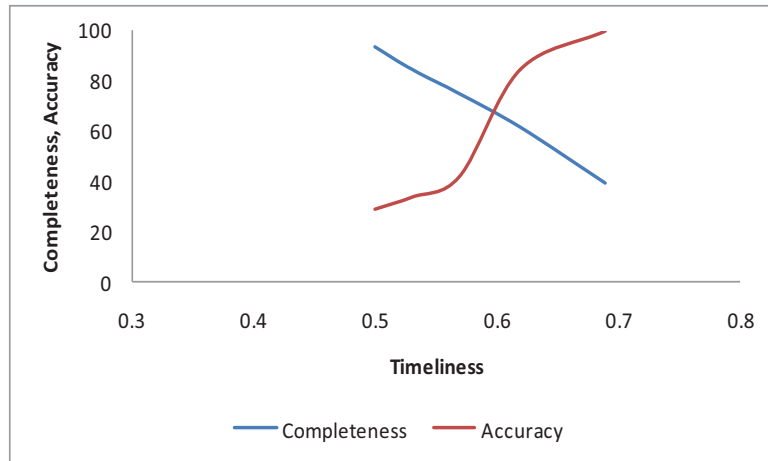
**Figure 5.23: Outbound Data Quality with Timeliness Graph for IMS4.2**

There is no difference between IMS4.1 and IMS4.3 except the insertion of the volume of data in the refreshment process. In the IMS4.3, low volume of data (60000) is used. Therefore, experiment 2 will be used for measuring the DQ of IMS4.3. The experimental result of IMS4.3 is given in Table 5.18.



**Table 5.18: Outbound Data Quality with Timeliness for IMS4.3**

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-2248360	11:27:05	11:31:24	259179	836063	.69	39.38	100.00
2	0-2722650	11:27:05	11:32:22	317703	836063	.62	61.40	84.56
3	0-3159300	11:27:05	11:33:04	359503	836063	.57	74.82	41.74
4	0-3320220	11:27:05	11:33:37	392890	836063	.53	84.20	33.48
5	0-3625700	11:27:05	11:34:03	418560	836063	.50	93.67	28.65

**Figure 5.24: Outbound Data Quality with Timeliness Graph for IMS4.3**

### 5.3 Summary

In this chapter, it is theoretically shown that DQ is changed with timeliness in heterogeneous IMS. In addition, the change of DQ with timeliness for heterogeneous IMS is shown experimentally. Therefore, it can be said that DQ changes with timeliness in the IMS. Moreover, there is a variation in the change. Therefore, the reasons for the change of DQ in IMS are also found from both the experiment and theory. Chapter 6 will discuss the improvement of DQ with timeliness by comparing the variation of change in DQ in heterogeneous IMS.



## **6 Discussion**

---

### **6.1 Introduction**

The research findings of this thesis are discussed in this chapter by comparing the DQ of heterogeneous IMS. Comparison of DQ of heterogeneous IMS was conducted by both theoretical analysis and by experimental results.

### **6.2 Data Quality Discussion for the Theoretical Analysis of Heterogeneous IMS**

DQ among heterogeneous types of DSS oriented IMS is analysed from a theoretical point of view. The factors of IMS such as fixed (high or low) machine capacity, fixed (high or low) volume of data, similar change frequency of data and continuous refreshment frequency method are considered. The heterogeneous DSSs are representing the factors such as the execution method of the refreshment and query function, the execution method of the tasks of refreshment function and the execution of the number of tasks for refreshment function. It needs time for refreshing data for the factors of DSS in IMS. A large volume of data could be in the DSS of IMS. Therefore, it needs more time for responding to the query request if both refreshment process and query request need to be executed in a single DSS at the same time. The time constraint query method is considered for responding to the query on time for each type of DSS. As a result, the execution method of refreshment and the query function is not important to ensure the query response on time. Therefore, the execution method of tasks of refreshment function and the execution of the number of tasks for refreshment



function play the role of changing DQ in IMS. Hence, it has been shown that DQ changes for the factors of heterogeneous DSS in IMS.

**Table 6.1: DQ Comparison of Heterogeneous DSS Oriented IMS in Theoretical Aspect**

<b>DSS Oriented Heterogeneous IMS</b>	<b>Execution Method of Tasks of Refreshment Function</b>	<b>Number of Tasks for Refreshment Function</b>	<b>Refreshment Period</b>	<b>Data Quality Status</b>	<b>Data Quality Performance</b>
Single DSS Oriented IMS	Sequential	$L + Ix$	$n(L+Ix)$	Changed	Better Than Cluster DSS
Cluster DSS Oriented IMS	Sequential	$L + Ix + P$	$n(L + Ix) + mP$	Changed	-----
2-DSS Oriented IMS	-----	$L$	$nL$	Changed	Better Than 3-DSS
3-DSS Oriented IMS	Simultaneous	$L + Ix$	$(L + nIx)$	Changed	Better Than Single DSS

Furthermore, it is shown in the Table 6.1 that there is a variation in the refreshment period in heterogeneous DSS oriented IMS for the execution method of tasks of refreshment function and the execution of the number of tasks for refreshment function. Therefore, there is a variation of DQ in the DSS oriented IMS. Only loading task is executed in the 2-DSS oriented IMS for the refreshment process. As a result, the refreshment period of data will be short in 2-DSS oriented IMS. Therefore, it provides better quality data than the other DSSs. Single and 3-DSS oriented IMS execute both loading and indexing task for the refreshment process. However, these tasks execute sequentially and simultaneously in single and 3-DSS oriented IMS respectively. In the single DSS, after loading a piece of data, indexing of that data is done. Each time a new piece of data is loaded, indexing of the existing data needs to be updated for the new piece of data. Hence, it needs more time for an indexing task than loading a data task.



Therefore, it needs more time for the refreshment processing of data. On the other hand, in 3-DSS, loading and indexing tasks execute simultaneously. Furthermore, it needs more time for an indexing task than loading a data task. Therefore, more data are loaded in DSS at the indexing period. As a result, like the single DSS, the indexing of 3-DSS does not need to update after loading a single piece of data. Rather, indexing of data is updated after multiple pieces of data loading for the simultaneous execution of the tasks of the refreshment function. Therefore, the refreshment period of 3-DSS oriented IMS is shorter than the single DSS oriented IMS. As a result, 3-DSS oriented IMS provides better quality data than the single DSS oriented IMS. Moreover, cluster DSS oriented IMS execute loading, indexing and propagation task for the refreshment process and execute sequentially. Therefore, the refreshment period of this DSS is longer than the single or other DSS oriented IMS. Hence, it provides worst quality data than among DSS oriented IMS. DQ comparisons for the factors of the DSS and other factors of IMS are discussed from an experimental point of view in the section 6.3 as well.

### **6.3 Data Quality Discussion for the Experimental Results of Heterogeneous Simulated IMS**

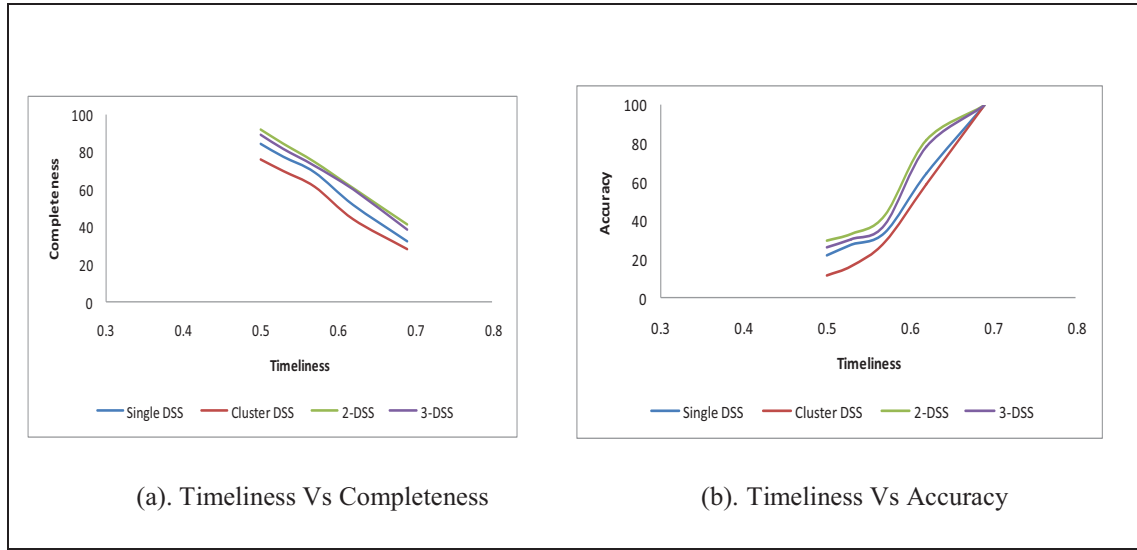
Theoretically, multiple factors of the heterogeneous IMS are considered together for showing the change and improvement of DQ with timeliness in the IMS. There are two dimensions of each individual factor of the heterogeneous IMS. Heterogeneous IMS are constructed by these two dimensions of each individual factor as shown in chapter 3. Therefore, DQ comparison between dimensions of each individual factors are done with the experimental results in chapter 5, by showing the change and improvement of DQ with timeliness in heterogeneous IMS.



### **6.3.1 Data Quality Comparison of DSS Oriented Heterogeneous IMS**

IMS1.1, IMS2, IMS3 and IMS4.1 are used for comparing the DQ of heterogeneous DSS oriented IMS. The capacity of the machine was high in these IMSs. The change frequency of data and the volume of data were non-frequent and high respectively in these IMSs. Continuous refreshment frequency method is used for refreshment processing for each type of IMS. Further, no overhead tasks were included to these IMSs. In these IMSs, factors of the DSS are varied. Execution method of refreshment and query function is one of the factors of DSS. It is executed both sequentially and simultaneously in heterogeneous IMS. Non-time constraint query method is used for query processing for each type of IMS. A query needs long time to respond if the volume of data in the DSS exceeds a certain level. The volume of data in the heterogeneous DSS oriented IMS does not exceed the certain level. As a result, the query response is instantaneous in the experiment of each type of IMS. Therefore, DQ varies for the refreshment period. The refreshment period varies in these IMSs for the execution method of tasks of refreshment function and the execution of the number of tasks for refreshment function. DQ comparison of DSS oriented heterogeneous IMS is shown in Figure 6.1. Refreshment period of 2-DSS oriented IMS is low. On the other hand, refreshment period of cluster DSS oriented IMS is high. Further, refreshment period of 3-DSS oriented IMS is less than the refreshment period of single DSS oriented IMS. It is shown in Figure 6.1 that 2-DSS oriented IMS and the cluster DSS oriented IMS provide good and poor quality data respectively than the other DSS. Furthermore, 3-DSS oriented IMS provides better quality data than the single DSS oriented IMS. The DQ scenario of this graph is also shown in Table 6.2.





**Figure 6.1: Data Quality Comparison of DSS Oriented Heterogeneous IMS**

**Table 6.2: Data Quality Scenario of Heterogeneous DSS Oriented IMS**

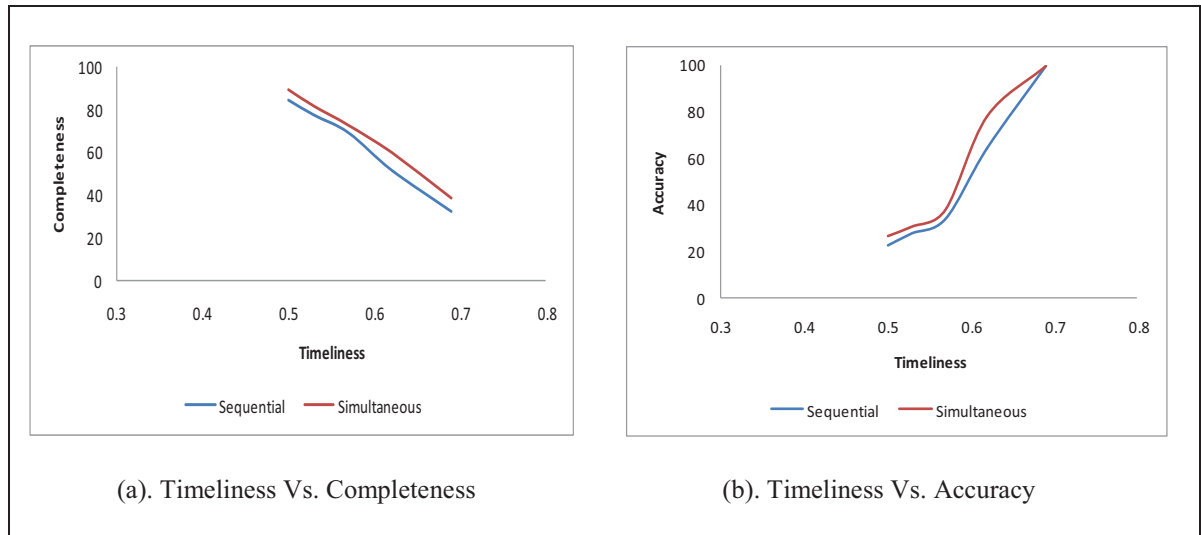
DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
Single DSS Oriented IMS	Sequential	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Cluster DSS Oriented IMS
Cluster DSS Oriented IMS	Sequential	L + Ix + P	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	-----
2-DSS Oriented IMS	-----	L	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than 3-DSS Oriented IMS
3-DSS Oriented IMS	Simultaneous	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Single DSS Oriented IMS

### 6.3.2 Data Quality with Timeliness for the Execution Method of Refreshment Function in IMS

In Figure 6.2, comparison of DQ between sequential and simultaneous execution method of refreshment function are shown. It is seen that DQ (completeness and accuracy) changes with timeliness for sequential and simultaneous execution method of



refreshment function oriented IMS. However, simultaneous execution method oriented IMS provide more improved DQ than the sequential execution method oriented IMS summarised in Table 6.3.



**Figure 6.2: Comparison of Data Quality with Timeliness for the Execution Method of Refreshment Function in IMS**

**Table 6.3: Data Quality Scenario for the Execution Method of Refreshment Function in IMS**

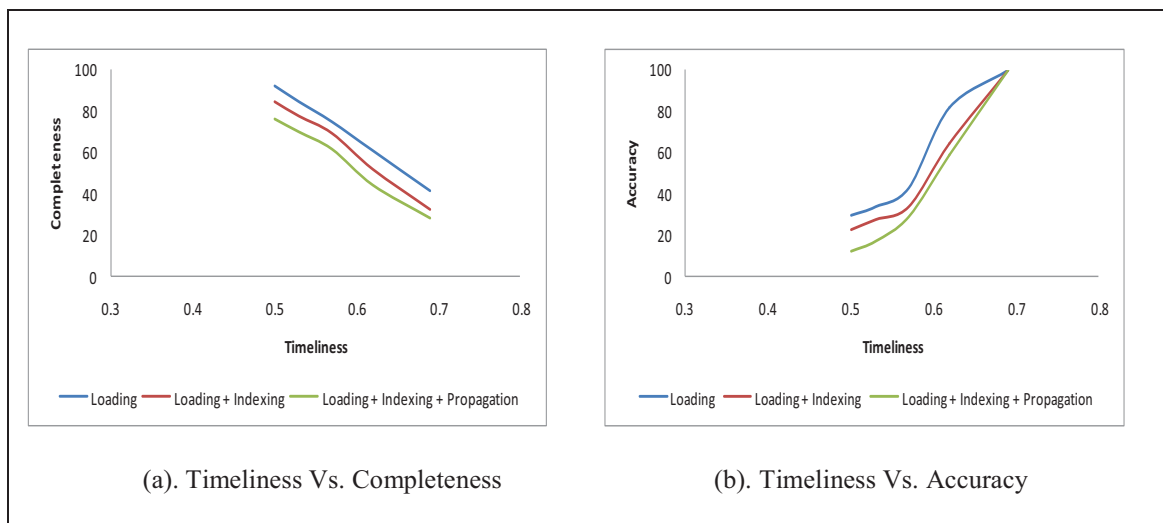
DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
3-DSS Oriented IMS	Simultaneous	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Quality Data Than the Sequential Method
Single DSS Oriented IMS	Sequential	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	-----

### 6.3.3 Data Quality with Timeliness for the Execution of Number of Tasks for the Refreshment Function of IMS

The comparison of DQ among the execution of the number of tasks for the refreshment function in the IMS is shown in Figure 6.3. It can be seen that DQ (completeness and



accuracy) is changed with timeliness for the execution of the number of tasks for the refreshment function in the IMS. However, the IMS that executes only the loading task provides more improved DQ than the IMS that execute both loading and indexing tasks. Further, the IMS that executes loading, indexing and propagation tasks provides poorer DQ than the IMS that execute both loading and indexing tasks as summarised in Table 6.4.



**Figure 6.3: Comparison of Data Quality with Timeliness for the Execution of Number of Tasks for Refreshment Function in IMS**

**Table 6.4: Data Quality Scenario for the Execution of Number of Tasks for Refreshment Function in Heterogeneous IMS**

DSS Oriented Heterogeneous IMS	Execution Method of Tasks for Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
2-DSS Oriented IMS	-----	L	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Single DSS Oriented IMS
Single DSS Oriented IMS	Sequential	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Cluster DSS Oriented IMS
Cluster DSS Oriented IMS	Sequential	L + Ix + P	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	-----



### 6.3.4 Data Quality with Timeliness for the Machine Capacity in IMS

In the Figure 6.4, the comparison of DQ between different capacity machines is shown. It is seen that DQ (completeness and accuracy) changes with timeliness for both high and low capacity machine oriented IMS. However, high capacity machine oriented IMS provides more improved DQ than the low capacity machine oriented IMS as shown in Table 6.5.

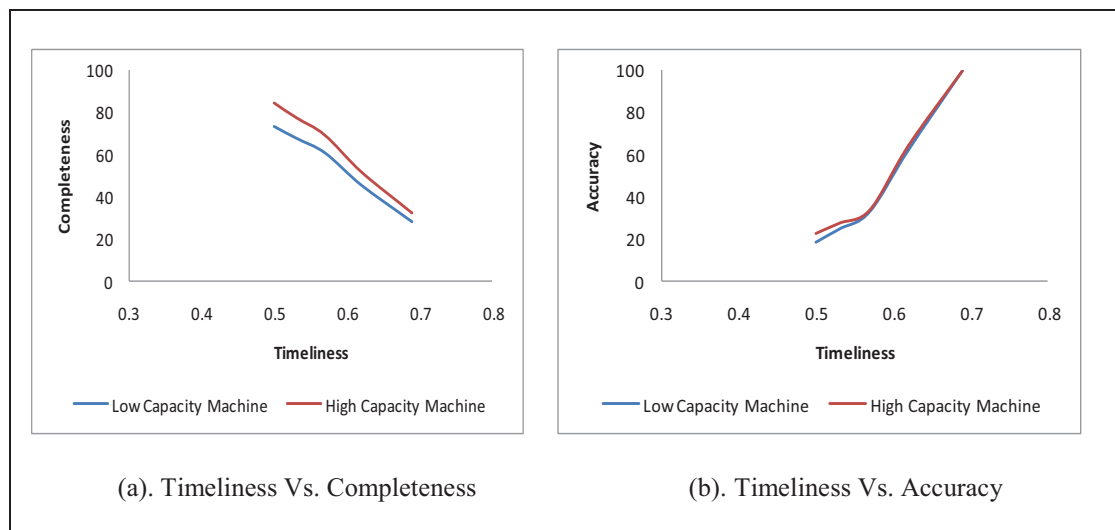


Figure 6.4: Comparison of Data Quality with Timeliness for the Machine Capacity in IMS

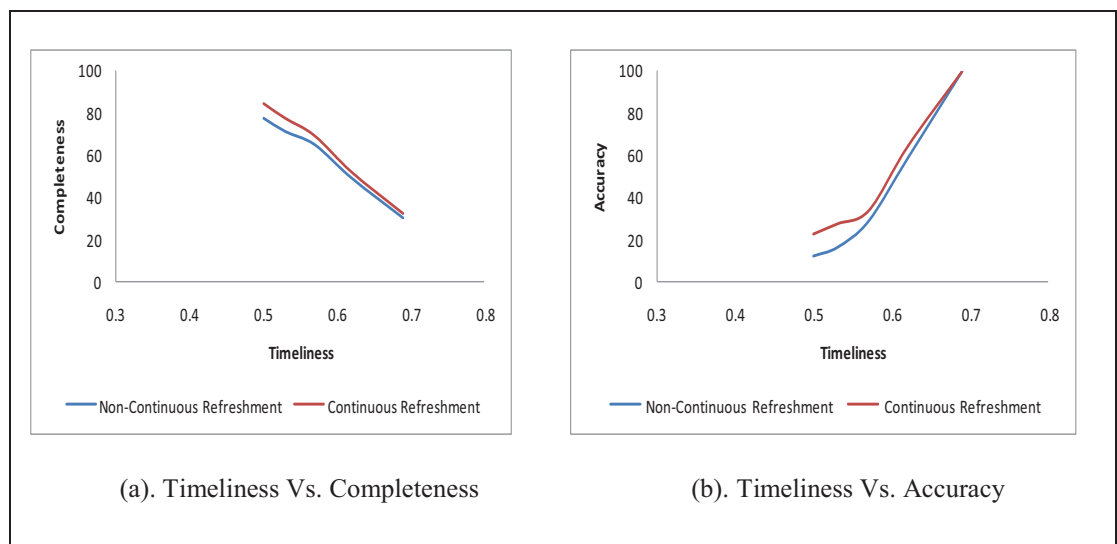
Table 6.5: Data Quality Scenario for the Machine Capacity in Heterogeneous IMS

DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
Single DSS Oriented IMS	Sequential	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Low Machine Capacity
	Sequential	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	Low	High	Non-Frequent	Changed	-----



### 6.3.5 Data Quality with Timeliness for Refreshment Frequency Method in IMS

Comparison of DQ between continuous and non-continuous refreshment frequency method is shown in Figure 6.5. It is seen that DQ (completeness and accuracy) is changed with timeliness for the continuous and the non-continuous refreshment frequency method oriented IMS. However, continuous refreshment frequency method oriented IMS provides more improved DQ than the non-continuous refreshment frequency method oriented IMS as summarised in Table 6.6.



**Figure 6.5: Comparison of Data Quality with Timeliness for the Refreshment Frequency Method in IMS**

**Table 6.6: Data Quality Scenario for the Refreshment Frequency Method in IMS**

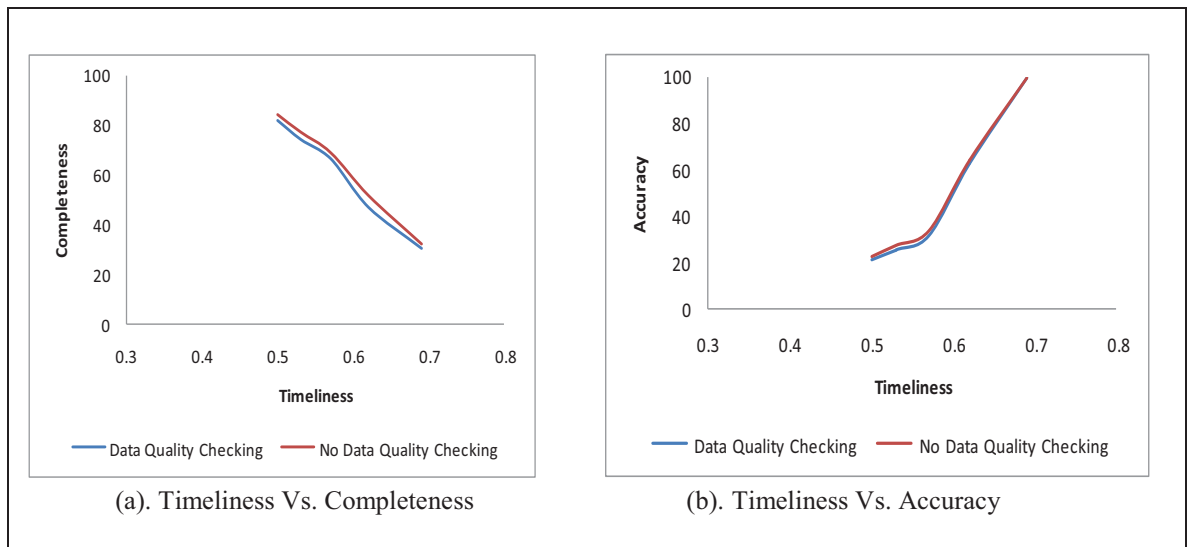
DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
Single DSS Oriented IMS	Sequential	$L + Ix$	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Non-Continuous Refreshment
	Sequential	$L + Ix$	No DQ Checking	Non-Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	-----



### 6.3.6 Data Quality with Timeliness for the Overhead Task of Refreshment

#### Function in IMS

In the Figure 6.6, the comparison of DQ between the inbound DQ checking and without inbound DQ checking oriented IMS is shown. The change of DQ (completeness and accuracy) with timeliness for the inbound DQ checking and without inbound DQ checking oriented IMS is seen in this figure. However, for the time consumption of DQ checking of inbound data, the inbound DQ checking oriented IMS provides poorer DQ than the inbound DQ checking oriented IMS as summarised in Table 6.7.



**Figure 6.6: Comparison of Data Quality with Timeliness for the Overhead Task of Refreshment Function in IMS**

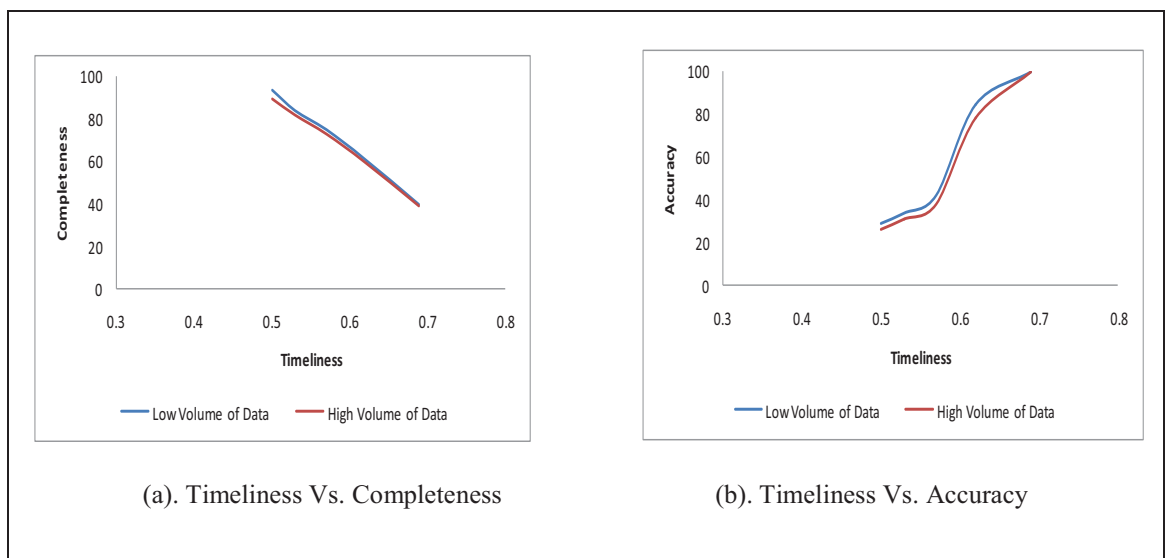


**Table 6.7: Data Quality Scenario for the Overhead Task of Refreshment Function in IMS**

DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
Single DSS Oriented IMS	Sequential	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than DQ Checking
	Sequential	L + Ix	DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	-----

### 6.3.7 Data Quality with Timeliness for the Volume of Data in IMS

In Figure 6.7, the comparison of DQ between high and low volume of data is shown. It is seen that DQ (completeness and accuracy) changes with timeliness for flow of high and low volume of data oriented IMS. However, the flow of the low volume of data oriented IMS provides more improved DQ than the flow of high volume of data oriented IMS as is summarised in Table 6.8.



**Figure 6.7: Comparison of Data Quality with Timeliness for the Volume of Data in IMS**

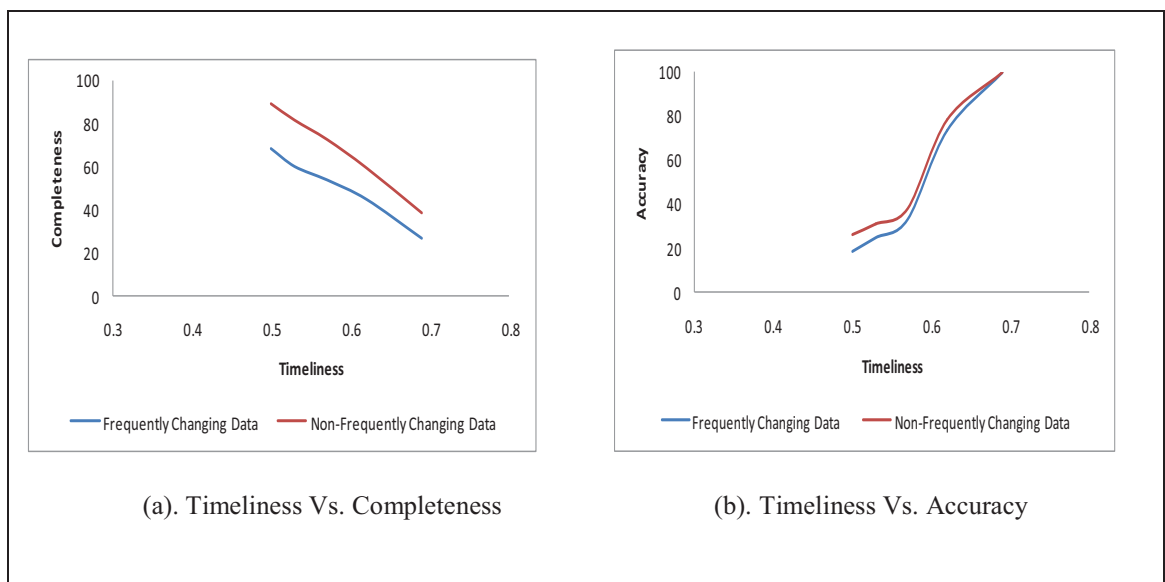


**Table 6.8: Data Quality Scenario for the Volume of Data in IMS**

DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
3-DSS Oriented IMS	Simultaneous	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	Low	Non-Frequent	Changed	Better Than High Volume of Data
	Simultaneous	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	-----

### 6.3.8 Data Quality with Timeliness for the Change Frequency of Data in IMS

Comparison of DQ between frequent and non-frequent change frequency of data is shown in Figure 6.8. It is seen that DQ (completeness and accuracy) changes with timeliness for the different change frequency of data oriented IMS. However, frequent change frequency data oriented IMS provides poorer DQ than the non-frequent change frequency data oriented IMS. This is summarised in Table 6.9.



**Figure 6.8: Comparison of Data Quality with Timeliness for the Change Frequency of Data in IMS**



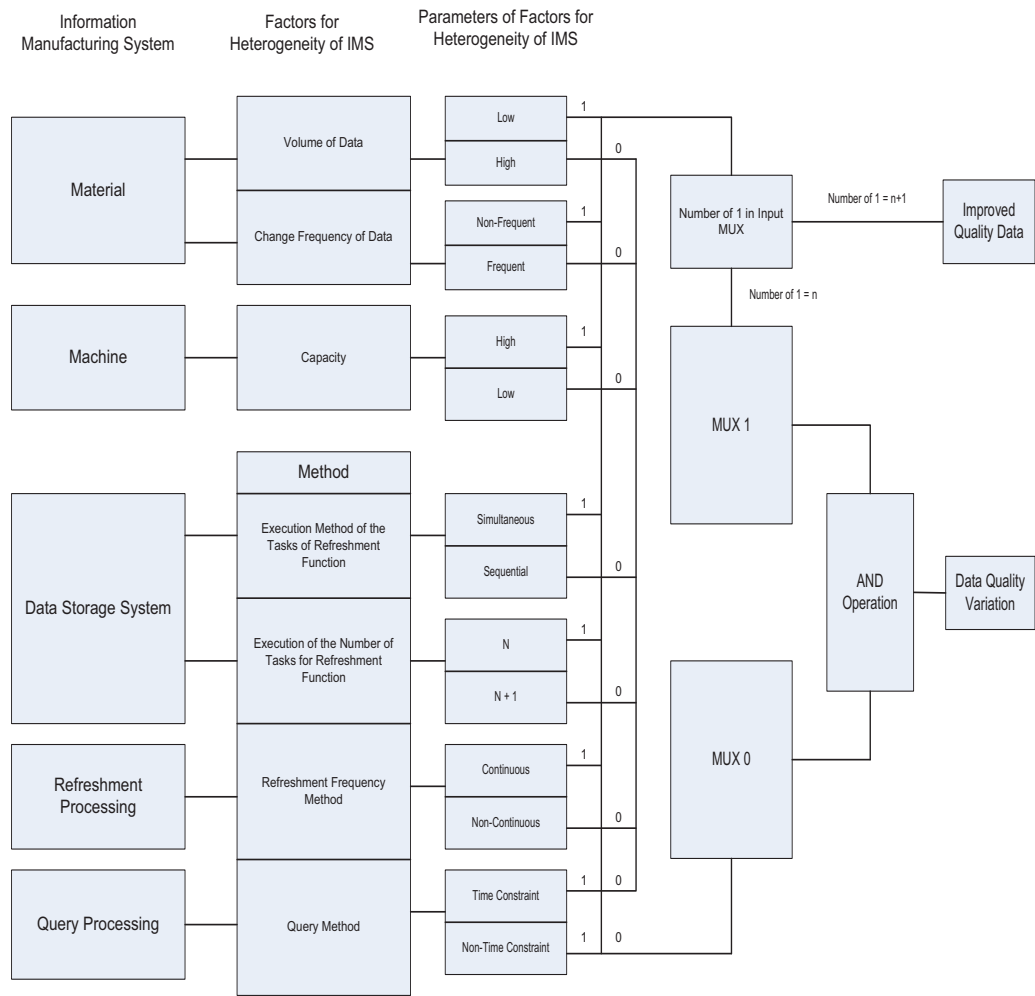
**Table 6.9: Data Quality Scenario for the Change Frequency of Data in IMS**

DSS Oriented Heterogeneous IMS	Execution Method of Tasks of Refreshment Function	Number of Tasks for Refreshment Function	Overhead Task of Refreshment Function	Refreshment Frequency Method	Query Method	Machine Capacity	Volume of Data	Change Frequency of Data	Data Quality Status	Data Quality Performance
3-DSS Oriented IMS	Simultaneous	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Non-Frequent	Changed	Better Than Frequent Change Frequency of Data
	Simultaneous	L + Ix	No DQ Checking	Continuous	Non-Time Constraint	High	High	Frequent	Changed	-----

## 6.4 Summary

It has been shown theoretically and experimentally that DQ changes with timeliness in heterogeneous IMS. However, there is a variation of the change of DQ in the heterogeneous IMS. The variation of the change of DQ occurs for the parameters ((low, high), (frequent, non-frequent) etc.) of individual factor of heterogeneous IMS. Therefore, the parameters of the factors of the heterogeneous IMS for the improvement of DQ with timeliness are identified by comparing the DQ of heterogeneous IMS.





**Figure 6.9: Scenario of Data Quality Variation & Improved Quality Data in IMS**

The scenario of DQ variation and the improved DQ in the IMS are shown in Figure 6.9. The specific research findings are indicated by this figure and are highlighted in the conclusions.



## 7 Conclusion

---

### 7.1 Research Findings

DQ in IMS is critical if organizations are to use them successfully. This work has looked at the measurement and improvement of DQ in IMS with specific reference to timeliness. A suitable measurement was first established, and then a scheme to ensure that quality could be improved was developed. The outcome of this work is novel in the following respects.

- The different parameters of the factors of heterogeneous IMS, which affect the variation of DQ with respect to timeliness, have been identified.
- An IMS quality system has been developed to understand the relationship between different parameters and the variation of DQ.
- The system is constructed to ensure that improvement of the parameter of individual factor improves the global quality of the data.

To the best of my knowledge, no such type of research has been done in the field of DQ or the IMS.

### 7.2 Limitations of This Research

There were limitations of the DQ assessment tool. It was not possible to use each type of manipulation operation in the experiments for each type of simulated IMS. Basically, simultaneous manipulation operations were not done in the simulated IMS experiment. Only single manipulation operations (insert, delete and update) are used in the experiments. The insert operation has been used in each type of simulated IMS. Whereas the delete and update operations are used only for the cluster DSS oriented



simulated IMS. Therefore, it was not possible to show the change of DQ for each manipulation operation done in an IMS of a real world organization.

### **7.3 Future Work**

In future, simultaneous manipulation operations will be done to find the effect on DQ in the IMS with timeliness DQ dimension. Therefore, DQ assessment tool will be improved. Further, an experiment for the partitioned DSS oriented IMS will also be done in future work.

In this research, it is shown that DQ changes with timeliness in the heterogeneous IMS. This research was undertaken from a DQ point of view. Among the used DSS oriented IMS, it is seen that 2-DSS oriented IMS could provide the best DQ. However, it cannot provide information support seamlessly to the user. As a result, it can provide DQ only for a certain period of time. It is also noticeable that 3-DSS oriented IMS has the ability to provide seamless information support to the user. It is also shown in the conference paper added to the end of this thesis. Therefore, future work will be to develop this 3-DSS. This future work will be done from the DSS point of view. In the development of 3-DSS, DQ level with timeliness will be like 2-DSS oriented IMS, but continuous. Partitioning is done in the DSS of the IMS for performance and availability of data in the IMS. The current partitioning procedure in a real world organization is static and pre-assigned. Therefore, it is not possible to create a new partition for increasing the performance of the IMS at the time of manipulation of data in the system. As a result, a dynamic partitioning procedure could work in this 3-DSS. It will not be a pre-assigned partition. This partition will be created only for the deterioration of the performance of the IMS.



## References

---

- Ballou, D.P., Wang, R.Y., Pazer, H.L. and Tayi, G.K., (1998), Modeling Information Manufacturing System to Determine Information Product Quality. *Management Science*. 44(4), pp. 462-484.
- Ballou, D.P. and Pazer, H.L., (1985), Modeling data and process quality in multi-input, multi-output information systems. *Management Science*, 31(2), pp. 150–162.
- Batini, C. and Scannapieco, M., (2006), Data Quality: Concepts, Methodologies and Techniques. *Publisher: Springer, Berlin, Germany*.
- Bobrowski, M., Marre, M. and Yankelevich, D., (1999), A Homogeneous Framework to Measure Data Quality. *Proceedings of International Conference on Information Quality*, pp. 115-124 Cambridge, MA.
- Bouzeghoub, M., Fabret, F. and Matulovic-Broqué, M., (1999), Modeling Data Warehouse Refreshment Process as a Workflow Application. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99)*, pp. 6.1-6.12.
- Bovee, M., Srivastava, R.P. and Mak, B., (2001), A conceptual framework and belief-function approach to assessing overall information quality. *Proceedings of the Sixth International Conference on Information Quality*.
- Bruckner, R.M., List, B., Schiefer, J. and Tjoa, A.M., (2001), Modeling Temporal Consistency in Data Warehouses. *Proceedings of the 12th International Workshop on Database and Expert Systems Applications*, pp. 901-905.
- Cappiello, C., Francalanci, C. and Pernici, B., (2003), Time-Related Factors of Data Quality in Multichannel Information Systems. *Journal of Management Information Systems*, 20(3), pp. 71-92.
- Cappiello, C., Francalanci, C. and Pernici B., (2005), A Self-monitoring System to Satisfy Data Quality Requirements. *Springer Verlag*, 3761, pp. 1535-1552.



- Cappiello, C. and Helfert M., (2008). Analyzing Data Quality Trade-Offs in Data-Redundant Systems. *Interdisciplinary Aspects of Information Systems Studies, Physica-Verlag HD*, pp. 199-205.
- Cappiello, C., Francalanci, C., Pernici, B., Plebani, P. and Scannapieco, M., (2003), Data quality assurance in cooperative information systems: a multi-dimension quality certificate. *Proceedings of the International Workshop on Data Quality in Cooperative Information Systems*.
- Catarci, T. and Scannapieco M., (2002), Data Quality under the Computer Science Perspective. *Journal of Archivi & Computer*.V-2.
- <http://www.dis.uniroma1.it/~monscan/ResearchActivity/Articoli/ArchiviComputer2002.pdf>. (Accessed 11 October 2009).
- Chaudhuri, S. and Dayal, U., (1997), An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD Record*, 26(1), pp. 65-74.
- Cochinwala, M., Kurien, V., Lalk, G. and Shasha, D., (2001), Efficient Data Reconciliation. *Information Sciences*, 137(1-4), pp. 1-15.
- Dong, C., Sampaio, M., and Sampaio, F., (2006), Expressing and Processing Timeliness Quality Aware Queries: The DQ<sup>2</sup>L Approach. *Springer Verlag*, 4231, pp. 382-391.
- Fisher, C.W. and Kingma, B.R., (2001), Criticality of data quality as exemplified in two disasters. *Information & Management*, 39(2), pp. 109-116.
- Ge, M. and Helfert, M., (2008), Data and Information Quality Assessment in Information Manufacturing Systems. *Springer Verlag*, 7, pp. 380-389.
- Ge, M. and Helfert, M. (2008), Modeling data quality in information chain, *International Conference on Business Innovation and Information Technology*, Dublin, Ireland.



- Ha, S.H. and Park, S.C., (1998), Data Modeling For Improving Performance Of Data Mart. *International Conference on Engineering and Technology Management, IEEE*, pp. 436-441.
- Hu, Y., Sundara, S. and Srinivasan, J., (2007), Supporting Time-Constrained SQL Queries in Oracle. *Proceedings of the 33rd international conference on Very large databases*, pp. 1207-1218.
- Huang, K.T., Lee, Y., Wang, R.Y., (1999), Quality information and knowledge management. *Publisher: Prentice Hall. New Jersey, USA*.
- Hsien, N.C., Chiang, D.A. and Chiang, R.C.T., (2001), Measuring the quality of queries in the fuzzy relational databases. *International Journal of Intelligent Systems*, 16(2), pp. 191-208.
- Inmon, W. H., Terdeman, R. H., Norris-Montanari, J. and Meers, D., (2001), Data Warehousing for E-Business. *Publisher: J. Wiley & Sons*.
- Islam, M.S. and Young, P., (2013), Modeling a Data Storage System (DSS) for Seamless Real-Time Information Support from Information Manufacturing System (IMS). *The Fifth International Conference on Advances in Databases, Knowledge and Data Applications (DBKDA)*, pp. 134-142.
- Jarke, M., Jeusfeld, M.A., Quix, C. and Vassiliadis, P., (1999), Architecture and Quality in Data Warehouses: An Extended Repository Approach. *Information Systems*, 24(3), pp. 229-253.
- Jenkins, A.M. (1985), Research methodologies and MIS research. *In E. Mumford et al., Research Methods in Information Systems*, Amsterdam, Holland, pp. 103-117.
- Kaplan, B., Duchon, D., (1988), Combining qualitative and quantitative methods information systems research: a case study. *Journal of Management Information Systems Quarterly*, 12(4), pp. 571-586.
- Mannino, M.V. and Walter Z., (2006), A framework for data warehouse refresh policies. *Decision Support Systems*, 42(1), pp. 121-143 .



- Milano, D., Scannapieco, M. and Catarci, T., (2005), Peer-to-Peer Data Quality Improvement in the DaQuinCIS System. *Journal of Digital Information Management*, 3(3), pp. 156-165.
- Naumann F., (2002), Quality-Driven Query Answering for Integrated Information Systems, LNCS 2261.
- Naumann, F. and Rolker, C., (2000), Assessment methods for information quality criteria. *5th International Conference on Information Quality*, pp. 148-162.
- Orr, K., (1998), Data quality and systems theory. *Communications of the. ACM*, 41(2), pp. 66–71.
- Pape, C.L. and Gancarski, S., (2007), Replica Refresh Strategies in a Database Cluster. *Springer-Verlag, LNCS 4395*, pp. 679-691.
- Pernici, B. and Scannapieco, M., (2003), Data Quality in Web Information Systems, *Springer-Verlag, LNCS 2800*, pp. 48-68.
- Pierce, E.M., (2004), Assessing data quality with control matrices. *Communications of the ACM*, 47(2), pp. 82-86.
- Pipino, L.L., Lee, Y.W., Wang, R.Y., (2002), Data Quality Assessment. *Communications Of The ACM*, 45(4ve), pp. 211-218.
- Redman, T., (1996), Data quality for the information age. *Publisher: Artech House, Boston, Massachusetts, USA*.
- Redman, T., (1998), The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 41(2), pp. 79-82.
- Rizvi, S.S. and Chung, T.S., (2010), Flash Memory SSD based DBMS for Data Warehouses and Data Marts. *The 2<sup>nd</sup> International Conference on Computer and Automation Engineering, IEEE*, pp. 557-559.



- Santos, R.J. and Bernardino, J., (2008), Real-Time Data Warehouse Loading Methodology. *Proceedings of the International Database Engineering & Applications Symposium*, pp. 49-58.
- Segev, A., Weiping, F., (1990). Currency-Based Updates to Distributed Materialized Views. *Proceedings of the 6<sup>th</sup> International Conference on Data Engineering (ICDE)*, Los Angeles, USA.
- Shankaranarayanan, G., Wang R.Y. and Ziad, M., (2000), IP-MAP: Representing the Manufacture of an Information Product. *5<sup>th</sup> Conference on Information Quality*, MIT, pp. 1-16.
- Silberschatz, A., Korth H. F. and Sudarshan S., (1997), Database System Concepts. *Publisher: Mcgraw-Hill*.
- Theodoratos, D. and Bouzeghoub, M., (1999), Data Currency Quality Factors in DataWarehouse Design. *Proceedings of the International Workshop on Design and Management of Data Warehouses*, pp. 15.1-15.16.
- Ulosoy. O., (1995), Research Issues in Real-Time Database Systems, *Information Sciences*, pp. 124-148.
- Vassiliadis, P., Bouzeghoub, M., Quix, C., (2000), Towards Quality-Oriented Data Warehouse Usage and Evolution. *Journal of Information System*, 25(2), pp. 89-115.
- Vrbsky, S.V., (1996), A data model for approximate query processing of real-time databases. *ACM Data & Knowledge Engineering*, 41(1), pp. 79-102.
- Vrbsky, S.V. and Tomic, S., (1998), Satisfying timing constraints of real-time databases. *Journal of Systems and Software*, 21(1), pp. 63-73.
- Wand, Y. and Wang, R.Y., (1996), Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11), pp. 86-95.
- Wang, R.Y., and Strong, D.M., (1996), Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4), pp. 5-33.



Wang, R.Y., Ziad, M. and Lee Y.W., (2001), Data Quality. *Publisher: Kluwer Academic* .

Wang, R.Y., (1998), A product perspective on total data quality management. *Communications of the ACM*, 41(2), pp. 58-65.

Wang R. Y., Reddy M. and Gupta A., (1993), An object oriented implementation of quality data products. *Proceedings of Third Workshop on Information Technology and Systems*, pp. 670–677.

Wang, R.Y., Lee, Y.W., Strong, D.M. and Kahn, B.K., (2002), AIMQ: a methodology for information quality assessment. *Information & Management*. 40(2), pp. 133-146 .

Wang, R.Y., Storey V.C. and Firth, C.P., (1995), A framework for analysis of data quality research, *IEEE Transactions on Knowledge and Data Engineering*, 7(4), pp. 623-640.

Wang, R.Y., Lee, Y., Pipino, L. and Strong, D.M., (1998), Manage your information as a product, *Sloan Management Review*, 39(4), pp. 95-105.

Wang, R. Y., Kon, H. B. and Madnick, S. E., (1993), Data quality requirements analysis and modeling. *Ninth International Conference on Data Engineering*, pp. 670–677.

Wang, K. Q., Tong, S. R., Roucoules, L. and Eynard, B., (2008), Analysis of data quality and information quality problems in digital manufacturing. *Management of Innovation and Technology, 4<sup>th</sup> IEEE International Conference*, pp. 439-443.

Zdenek, K., Kamil, M., Petr, M. and Olga, S., (1998), On updating the data warehouse from multiple data sources. *Springer*, 1460, pp. 767-775.

Zhou, N. and Ding, Q., (2006), Open Real-time Database and it's Application in Dispatching Automation Systems. *International Conference on Power System Technology*, IEEE, pp. 1-6.

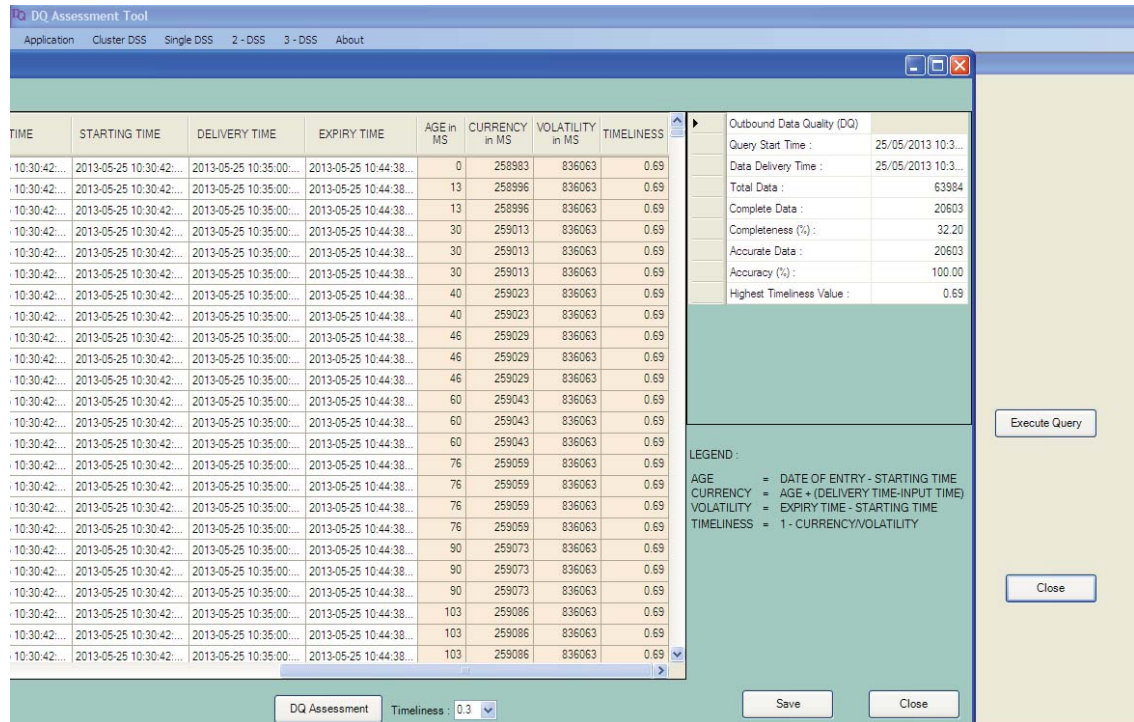
<http://www.tpc.org/tpch/> [Accessed 18<sup>th</sup> June 2013].

<http://www.tpc.org/tpch/spec/tpch2.14.4.pdf> [Accessed 18<sup>th</sup> June 2013].



# Appendix A

## Data Quality Experiment of Simulated Information Manufacturing System:



►	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 10:3...
	Data Delivery Time :	25/05/2013 10:3...
	Total Data :	63984
	Complete Data :	33208
	Completeness (%) :	51.90
	Accurate Data :	21440
	Accuracy (%) :	64.56
	Highest Timeliness Value :	0.62

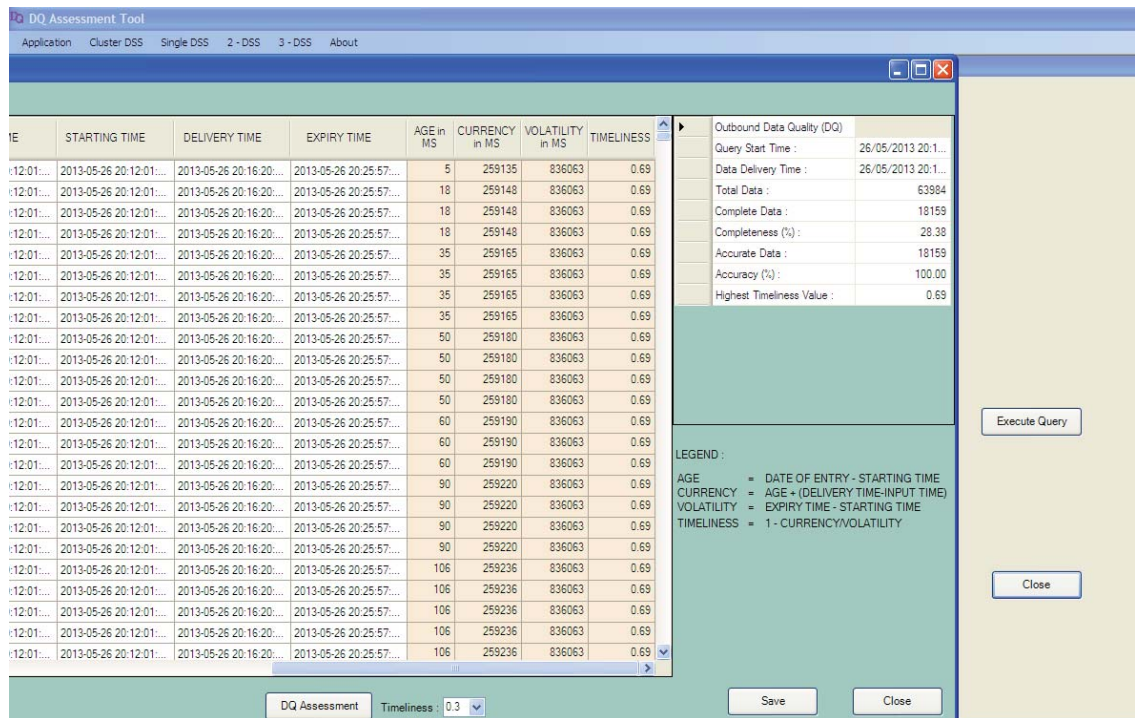
►	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 10:3...
	Data Delivery Time :	25/05/2013 10:3...
	Total Data :	63984
	Complete Data :	44341
	Completeness (%) :	69.30
	Accurate Data :	15019
	Accuracy (%) :	33.87
	Highest Timeliness Value :	0.57

►	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 10:3...
	Data Delivery Time :	25/05/2013 10:3...
	Total Data :	63984
	Complete Data :	49441
	Completeness (%) :	77.27
	Accurate Data :	13631
	Accuracy (%) :	27.57
	Highest Timeliness Value :	0.53

►	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 10:3...
	Data Delivery Time :	25/05/2013 10:3...
	Total Data :	63984
	Complete Data :	53939
	Completeness (%) :	84.30
	Accurate Data :	12137
	Accuracy (%) :	22.50
	Highest Timeliness Value :	0.50

Figure A1: Experimental Result of Single DSS Oriented IMS (IMS1.1)





Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 20:1...
Data Delivery Time :	26/05/2013 20:1...
Total Data :	63984
Complete Data :	28326
Completeness (%) :	44.27
Accurate Data :	16642
Accuracy (%) :	58.75
Highest Timeliness Value :	0.62

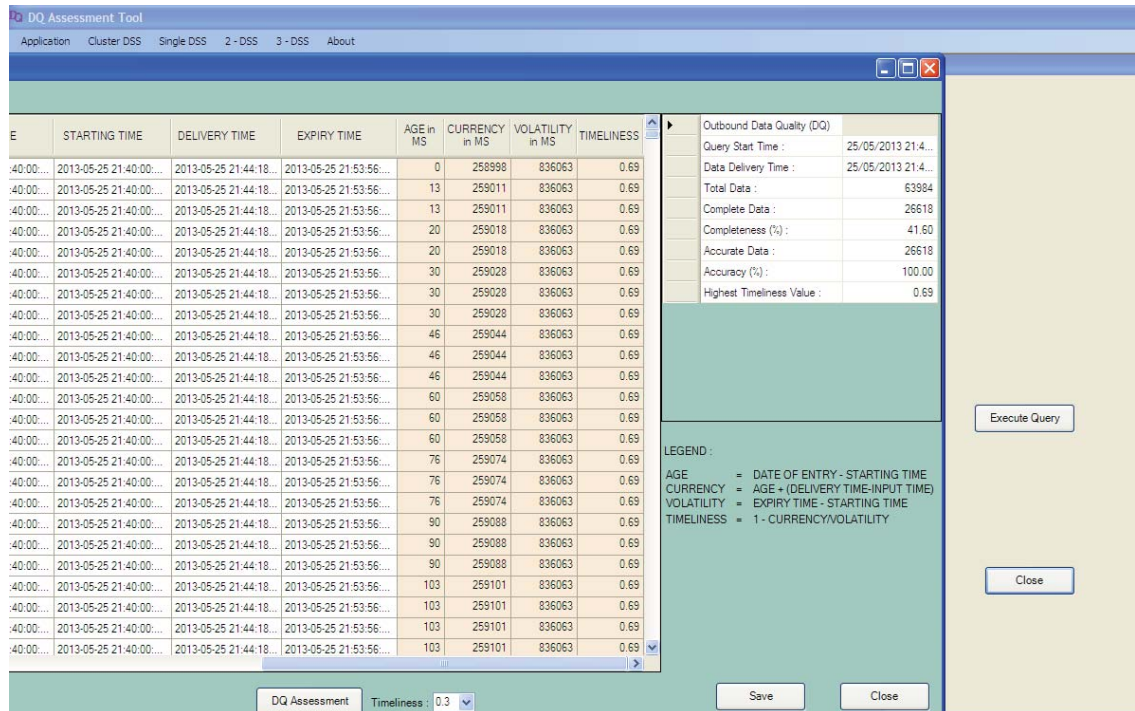
Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 20:1...
Data Delivery Time :	26/05/2013 20:1...
Total Data :	63984
Complete Data :	39363
Completeness (%) :	61.52
Accurate Data :	11337
Accuracy (%) :	28.80
Highest Timeliness Value :	0.57

Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 20:1...
Data Delivery Time :	26/05/2013 20:1...
Total Data :	63984
Complete Data :	44636
Completeness (%) :	69.76
Accurate Data :	7531
Accuracy (%) :	16.87
Highest Timeliness Value :	0.53

Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 20:1...
Data Delivery Time :	26/05/2013 20:1...
Total Data :	63984
Complete Data :	48852
Completeness (%) :	76.35
Accurate Data :	5887
Accuracy (%) :	12.05
Highest Timeliness Value :	0.50

Figure A2: Experimental Result of Cluster DSS Oriented IMS (IMS2)





▶	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 21:4...
	Data Delivery Time :	25/05/2013 21:4...
	Total Data :	63984
	Complete Data :	38890
	Completeness (%) :	60.78
	Accurate Data :	31719
	Accuracy (%) :	81.56
	Highest Timeliness Value :	0.62

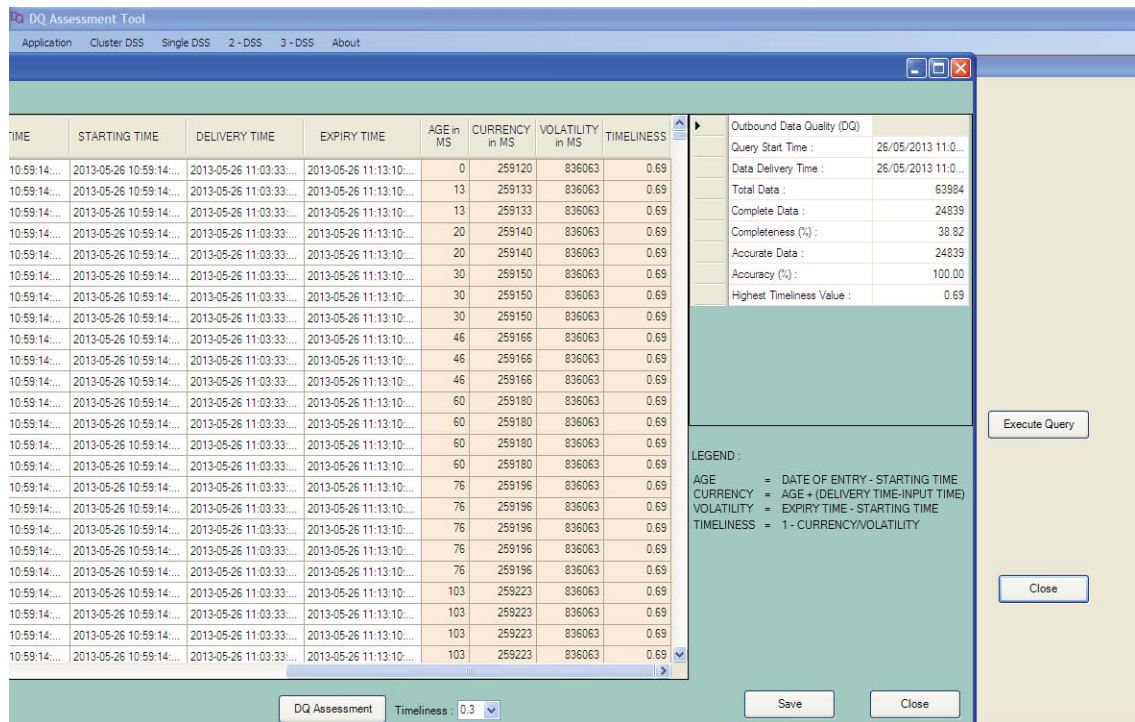
▶	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 21:4...
	Data Delivery Time :	25/05/2013 21:4...
	Total Data :	63984
	Complete Data :	48040
	Completeness (%) :	75.08
	Accurate Data :	20571
	Accuracy (%) :	42.82
	Highest Timeliness Value :	0.57

▶	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 21:4...
	Data Delivery Time :	25/05/2013 21:4...
	Total Data :	63984
	Complete Data :	54272
	Completeness (%) :	84.82
	Accurate Data :	18214
	Accuracy (%) :	33.56
	Highest Timeliness Value :	0.53

▶	Outbound Data Quality (DQ)	
	Query Start Time :	25/05/2013 21:4...
	Data Delivery Time :	25/05/2013 21:4...
	Total Data :	63984
	Complete Data :	59071
	Completeness (%) :	92.32
	Accurate Data :	17580
	Accuracy (%) :	29.76
	Highest Timeliness Value :	0.50

**Figure A3: Experimental Result of 2-DSS Oriented IMS (IMS3)**





Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 11:0...
Data Delivery Time :	26/05/2013 11:0...
Total Data :	63984
Complete Data :	38263
Completeness (%) :	59.80
Accurate Data :	29876
Accuracy (%) :	78.08
Highest Timeliness Value :	0.62

Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 11:0...
Data Delivery Time :	26/05/2013 11:0...
Total Data :	63984
Complete Data :	46683
Completeness (%) :	72.96
Accurate Data :	17693
Accuracy (%) :	37.90
Highest Timeliness Value :	0.57

Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 11:0...
Data Delivery Time :	26/05/2013 11:0...
Total Data :	63984
Complete Data :	52295
Completeness (%) :	81.73
Accurate Data :	16107
Accuracy (%) :	30.80
Highest Timeliness Value :	0.53

Outbound Data Quality (DQ)	
Query Start Time :	26/05/2013 11:0...
Data Delivery Time :	26/05/2013 11:0...
Total Data :	63984
Complete Data :	57202
Completeness (%) :	89.40
Accurate Data :	15045
Accuracy (%) :	26.30
Highest Timeliness Value :	0.50

Figure A4: Experimental Result of 3-DSS Oriented IMS (IMS4.1)



## Appendix B:

### **Published Paper:**

1. Islam, M.S. and Young, P., (2013), Modeling a Data Storage System (DSS) for Seamless Real-Time Information Support from Information Manufacturing System (IMS). The Fifth International Conference on Advances in Databases, Knowledge and Data Applications (DBKDA), pp. 134-142.
2. Islam, M.S. and Helfert, M., (2013), Data Quality Comparison between Highly Integrated Single and Three Data Storage System Oriented Information Manufacturing System. *The International Conference on E-Technologies and Business on the Web (EBW)*, IEEE, pp. 250-257.

These two papers are relevant to this thesis. Therefore, these papers are added to the following pages.



## Modeling a Data Storage System (DSS) for Seamless Real-Time Information Support from Information Manufacturing System (IMS)

Mohammad Shamsul Islam  
Faculty of Engineering & Computing  
Dublin City University (DCU)  
Dublin, Ireland  
Email: mohammad.islam6@dcu.ie

Paul Young  
Faculty of Engineering & Computing  
Dublin City University (DCU)  
Dublin, Ireland  
Email: paul.young@dcu.ie

**Abstract**—Nowadays, a large number of enterprises operate in a business time schedule of  $24 \times 7$ . These enterprises need to deliver information as fast as possible for information support. Therefore, the information manufacturing system of the enterprises should have the ability for seamless real-time information support. Data storage system in the information manufacturing system plays the role of providing non interrupted real-time information support. Therefore, 2-Data Storage System oriented information manufacturing system is developed for providing real-time information support. This 2-Data Storage System oriented information manufacturing system can provide real-time information support for a short period of time. However, it is not possible to provide seamless real-time information support by this information manufacturing system. Hence, modeling a data storage system for seamless real-time information support from the information manufacturing system is the purpose of this paper.

**Keywords**-data loading; indexing; query processing.

### I. INTRODUCTION

An IMS (Information Manufacturing System) is an information system that manufactures information from the raw data [23]. The most important component of the IMS is a DSS (Data Storage System). The DSS integrates multiple sources of the system and so contains raw data from multiple sources. Data come from multiple sources are processed by the refreshment function of the availability of data in the DSS. Available data in the DSS are then delivered as information by the execution of query function.

Traditionally, IMS works in the non real-time environment. Single or cluster (replication) DSS oriented IMS is used for providing information support for this non real-time environment. The DSS is updated periodically, typically in a daily, weekly or even monthly basis in the non real-time environment [24]. The DSS needs to update continuously for providing real-time information support with most recent data. Update is done with the refreshment function in the DSS. Continuous execution of the refreshment function (single DSS) and non simultaneous update (cluster DSS) can cause of the poor quality information support from the IMS [6]. More specifically,

the poor quality information support occurs for not executing the refreshment and query function simultaneously (single DSS) or the propagation delay for updating the DSS (cluster DSS) of IMS. Therefore, these DSS oriented IMS are not suitable for real-time information support.

Enterprises such as stock brokering, e-business, online telecommunication, health system and traffic systems need to deliver information as fast as possible to knowledge workers or decision-makers who make a decision in a real-time or near real-time environment, according to the new and most recent data captured by an organization's IMS [12]. Therefore, Santos and Berardino [18] as well as Hanson and Willshire [10] developed a 2-DSS oriented IMS for providing real-time information support. However, this 2-DSS oriented IMS cannot provide real-time information support seamlessly. Nowadays, some enterprises need to operate in a business time schedule of  $24 \times 7$  for providing information support in real-time environment. Therefore, the purpose of this research is for modeling a data storage system in the IMS that can provide non-interrupted real-time information support for the business time schedule of  $24 \times 7$ . The modeled data storage system is 3-DSS for serving the purpose.

The remaining part of this paper is organized as follows: Section 2 presents the related research of the data storage system. Section 3 describes the 3-DSS. Section 4 shows the regulating procedure of the tasks of the refreshment and query function in the 3-DSS. Section 5 presents the management of the system at the down period of principal 3-DSS and the execution of tasks for restarting the principal 3-DSS again in the system. Experimental evaluation as well as conclusion and future work are shown in Section 6 and Section 7 respectively.

### II. RELATED RESEARCH

So far, some researches have been done over DSS (DW, distributed DW, etc.). Bouzeghoub et al. [1] and Vavouras et al. [21] presents the modeling of the data warehouse refreshment process. They explain the difference between



loading and refreshment process to these papers. Analyzing the information manufacturing system of many organizations, Mannino and Walter [13] identify that timeliness and availability of data in the DSS are responsible for bad quality information in the IMS. They also find that the refresh period of a system influences the timeliness and availability of data. Theodoratus and Bouzeghoub [20] discuss the data currency quality factors in data warehouses and propose a DW design that considers these factors. An important issue for near real-time data integration is the accommodation of delays, which has been investigated for (business) transactions in temporal active databases in [17]. Vrbsky [22] developed a model to get approximate information from the IMS within a certain time in real-time environment. McCarthy and Risch [16] provide the data structure for execution of real-time queries. Capiello et al. [6] shows that multiple scattered DSS has the lowest degree of integration. It is evident that there is a data quality problem as a result of both long refresh period and propagation delay. On the other hand, single DSS in the IMS has the highest degree of integration, therefore, it does not make the data quality problem. Santos and Bernardino [18] present a table structure replication technique to ensure fresh decision support for the real-time or frequently changing data. The table structure replications have two tables, the permanent table and the temporary table. The data stored in a temporary table are transferred to a permanent table for the deterioration of the query response. At the time of transfer, no access is possible in the table of data storage for the information support. Hanson and Willshire [10] developed a faster data warehouse model providing an auxiliary structure for quick query response. This is also a 2-DSS oriented IMS. It has a temporary table as well where only data will be loaded and no administrative overhead such as indexing will be done. In this model storage capacity of temporary tables is limited as it is installed in the non-volatile NVRAM. After fulfilling the 95% of the temporary tables, data is transferred to the permanent table. Therefore, no data access will be possible in this period. As a result,  $24 \times 7$  services will be not possible with these 2-DSS oriented IMS seamlessly.

The data storage model of this research does not need to transfer data by pushing the system to offline or by stopping the system. Therefore, it will be possible to provide  $24 \times 7$  services seamlessly with this DSS model. Further, it will update the data in the DSS in real-time manner for providing the real-time information support.

### III. 3-DATA STORAGE SYSTEM (3-DSS)

According to [1][13][18], refreshment and query function execute in a data storage system of the IMS to make the data available and for the information support respectively.

**Refreshment Function:** This is a complex process comprising the tasks, such as data loading, indexing and propagation of data for synchronizing data in the information manufacturing system (IMS) [1][13][18].

**Data loading:** Key activities of data loading include extraction, transformation, integration, cleaning etc. Therefore, storage of manipulating [insert, update] data are to extract from the sources, then, transformed data if the source data are in the different format. After that, extracted and transformed data are to integrate and to clean for loading data in the data storage system [1][13][18].

**Indexing:** Update the index for newly loaded data or delete data to align the data in the data storage system [18]. Indexing determines the effective usability of data collected and aggregated from the sources and increases the performance of the data storage system for information support [1][13].

**Propagation of Data:** Data is propagated through the refreshment process for synchronizing the data of multiple DSSs of the system.

**Query Function:** This function of data storage system in IMS is done by the query processing task. A requested query of the user is processed in the data storage system for delivering the information to the user.

In the 3-DSS, three individual DSS are mutually interconnected with each other. The tasks of the refreshment and the query function of data storage system work simultaneously in three individual DSS. Therefore, data loading and indexing with updated data propagation task of the refreshment function work in two individual DSS of 3-DSS and another DSS of 3-DSS executes the task of the query function at the same time. After each successive period, the tasks of the refreshment and the query function of data storage system will interchange with cyclic order. As, propagation of manipulated data in the DSS is done simultaneously at the working period of the task of the functionalities, there may not have propagation delay. Therefore, 3-DSS will hold exact the same data. The 3-DSS is shown in Figure 1.

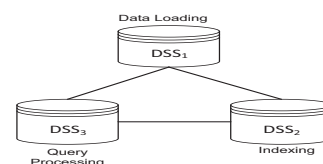


Figure 1. 3-DSS

In the following sub-sections, the execution process of 3-DSS, protocol of 3-DSS and the partitioning procedure of 3-DSS will be discussed.



### A. Execution Process of 3-DSS

Suppose, DB1, DB2 and DB3 is three individual DSS for 3-DSS. These three DSS are mutually interconnected with each other. Now, if DB1 store some data from operational data sources, DB2 and DB3 must have the same data. The tasks of the query and refreshment function work simultaneously in these three data storage systems. There must have a synchronization of starting and finishing time of the tasks of the query and refreshment function of these three data storage systems. Manipulated data from operational data sources will load into one database. At the same time, another database will do the indexing and updated data propagation task for synchronizing data with other two DSS and the third one will be used for query processing. The indexed and query processing database lead the process. When the indexing with the propagation of updated data and the query processing are finished, the tasks of the refreshment and query function of data storage system will interchange with cyclic order. The rotation algorithm for the interchanging process is given in the Figure 2.

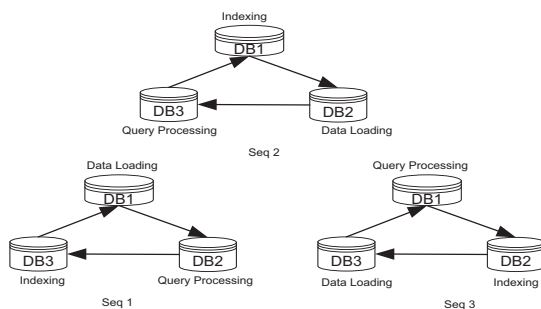


Figure 2. Rotation of the tasks of functionalities in 3-DSS

The algorithm for the rotation of function (Data Loading, Indexing and Query Processing) in data storage system is shown in Figure 3.

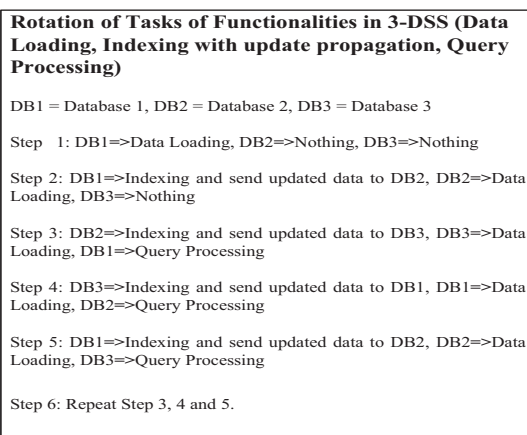


Figure 3. Algorithm for Rotation of Tasks of Functionalities in 3-DSS

In the algorithm, Steps 1 and 2 indicate the initialization of the system. Data come from multiple sources are loaded into DB1 in Step 1. DB1 executes the indexing task and send the updated data to DB2 and DB2 loads the updated data and source data simultaneously in Step 2. Each task of the functionalities of data storage system works simultaneously in Steps 3, 4 and 5. As, data have been loaded into DB2 in Step 2, DB2 is indexed in Step 3 and send the updated data to DB3. At the same time, DB3 loads the manipulated data including the updated data of DB2. Further, DB1 provides the information by processing the query request of the system in Step 3. Steps 4 and 5 will follow the same process but interchange the roles of each DB of the system. Therefore, Steps 3, 4 and 5 will continue repeatedly in the system.

### B. 3-DSS Protocol

A protocol is a set of rules for regulating a system [9]. 3-DSS is the data storage system where multiple tasks will execute simultaneously. Therefore, 3-DSS are to maintain a set of rule for avoiding the cumbersome operations of the 3-DSS in the IMS.

1. N, 2N, 3N,.....NN amount of manipulating new data will be loaded each time to be available in the DSS after completion of the task of the refreshment function.
2. Three individual DSS of the 3-DSS will be located contiguously in the same place.
3. Functionalities do not interchange if the indexing task is in progress. It means that functionalities do not interchange before the completion of indexing task.
4. Functionalities do not interchange if the propagation of update data from one DSS to another DSS is in progress. Otherwise, the propagated data receiver DSS can not load the all new data of the sender DSS.
5. Functionalities do not interchange if query processing task is in progress. Otherwise, the query can process an incomplete query result.
6. Throughput of the three interconnected network link of 3-DSS should be same.
7. Rotation of the tasks of functionalities will be clockwise cyclic order like the Figure 2.
8. Previous DSS of indexing DSS will always process the query and next DSS of indexing DSS always load the manipulated data. It is seen in Figure 2 that in every sequence (seq) previous DSS of indexing DSS process the query.



9. Previous DSS of query processing DSS will always load the manipulated data and next DSS of query processing will always indexed the loaded data of the DSS. It is seen in Figure 2 that in every sequence (seq) previous DSS of query processing DSS load the manipulated data.
10. Previous DSS of data loading will always indexed the loaded data of the DSS and next DSS of data loading DSS will always process the query. It is seen in Figure 2 that in every sequence (seq) previous DSS of data loading DSS indexed the loaded data.
11. Interchange of the tasks of functionalities will be done after the completion of the indexing and query processing tasks.

### C. 3-DSS Partitioning

Partitioning is the technique of fragmenting large relations (tables) into smaller ones. In the large DSS, (tables), if manipulation of data (insertion, update, delete) is done, it needs more time to rebuild the indices of the large DSS (tables). As a result, a requested query may not execute in time or may provide a poor query result. The partitioning of a large table can resolve this problem. In a partitioning DSS (tables), the problem that created at the time of index rebuilding for the manipulation of data is limited only in a particular partition. Therefore, except the partitions where data is being manipulated, other partitions of the DSS (tables) can provide the query result for the requested query as the index rebuilding process is limited to the certain partition. Additionally, it needs less time to rebuild the index than the non-partitioned DSS (tables) as the volume of data of each partition will be certain. There are two ways to partition a DSS: vertically and horizontally. Vertical partitioning involves splitting the attributes (columns) of a DSS (tables), placing them into two or more DSS (tables) linked by the DSS (tables) primary key. Horizontal partitioning involves splitting the tuples of a DSS (table), placing them into single or more DSS (tables) with the same structure. For keeping the certain volume of data in the DSS (table), horizontal partitioning will be used in the 3-DSS. There are two types of horizontal partitioning: primary and derived. Primary horizontal partition (HP) of a DSS (table) is performed using attributes defined on that DSS (table). On the other hand, derived horizontal partition is the fragmentation of a DSS (table) using the attributes defined on another DSS (tables) [4]. Horizontal partitioning for 3-DSS is discussed in below,

Let, F is the primary or fact table, D is the derived or dimensional table. Example of a primary (fact) relational table and derived (dimensional) table are depicted in Figure 4.

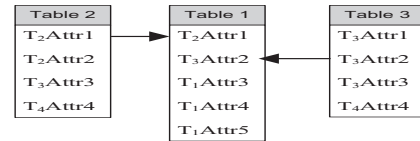


Figure 4. Primary (fact) relational table and derived (dimensional) table

In Figure 4, Table 2 and Table 3 is primary table. On the other hand, Table 1 is derived table. Fragmentation of Table 1 depends on Table 2 and Table 3. If primary Table 2 and Table 3 are manipulated, Table 1 must have to be manipulated. Therefore, tuple of Table 1, Table 2 and Table 3 will be horizontally partitioned simultaneously considering the instruction of the predicate.

A predicate is the Boolean expression over the attributes of a relational table and constants of the attribute's domains. Horizontal partitioning can be defined as a pair  $(T, \Phi)$ , where  $T$  is a relation and  $\Phi$  is a predicate. This predicate partitions  $T$  into at most 2 fragments with the same set of attributes. The first fragment includes all tuples of  $t$  of  $T$  which satisfy  $\Phi$ , i.e.,  $t = \Phi$ . The second fragment includes all tuples  $t$  of  $T$  which does not satisfy  $\Phi$ , i.e.,  $t \neq \Phi$ . It is possibly one of the fragments to be empty if all tuples of  $T$  either satisfy or do not satisfy  $\Phi$ .

Let  $\Phi = (\text{counted tuple} = N)$ , which results into fragment horizontally where tuple of a relational table will be counted. If the condition of the predicate  $\Phi$  is true, relational table will be fragmented into 2 partitions. The first partition will hold  $N$  numbers of the tuple. If manipulation of data in the information manufacturing system (IMS) is stopped after being partitioned, the second partition will remain empty. When manipulation of data in the IMS is continued and the total insertion of tuple reaches to  $N$  number, this partition will again fragment into two pieces. This partitioning process will continue as long as the information manufacturing system is not stopped. This single table partitioning process can be applied to the multiple relation tables of the DSS in the IMS. The horizontal Partitioning algorithm and the partitioning algorithm of the 3-DSS are given in Figure 5 and Figure 6, respectively.

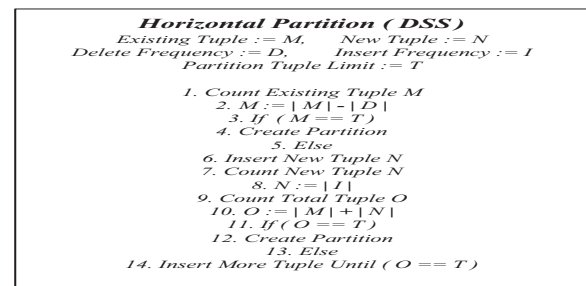


Figure 5. Horizontal Partitioning algorithm for 3-DSS



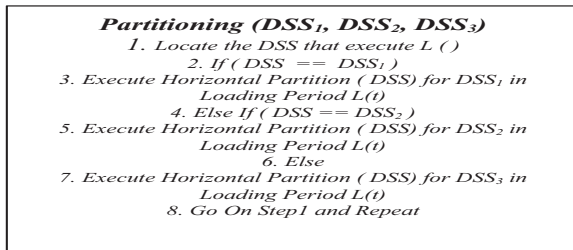


Figure 6. Partitioning algorithm for 3-DSS

Partitioning of DSS of the 3-DSS will be done by following the partitioning algorithm given in Figure 6. According to this algorithm, DSS will be located for the partitioning in the executing period of data loading of a particular DSS. Then, the horizontal partitioning algorithm will be applied for partitioning the DSS of the 3-DSS. The horizontal partitioning algorithm is shown in Figure 5. In this algorithm, Existing and new tuples have to calculate for the horizontal partitioning. Deleting of a tuple from DSS will detect the tuple from the existing tuple. On the other hand, insertion of the tuple will count as a new tuple. A DSS will be partitioned if total tuples of the DSS is reached in the partition limit  $N$ . Therefore, the existing tuple will be counted by deducting the deleted tuple from the DSS. Hence, the number of existing tuples will be counted for checking whether the existing tuple is in partition limit or not. If, it is in partition limit, the partition will be created. Otherwise, new tuple will be inserted in the DSS. Therefore, new tuple will be counted by the number of new insertions. After that, total tuple will be counted by adding existing tuple with new tuple. Henceforth, it will be checked for whether a counted number of total tuples are in partition limit or not. If total tuple equals to the partition limit, the partition will be created. Otherwise, more tuple has to be inserted until the total tuple reaches to the range of the partition limit of the tuple.

#### IV. REGULATING PROCEDURE OF THE TASKS OF REFRESHMENT & QUERY FUNCTION OF 3-DSS

3-DSS executes three individual DSS simultaneously in the information manufacturing system (IMS). The tasks of the functionalities of data storage system work in these three individual DSS of the 3-DSS. These tasks also need to interchange in the 3-DSS. Further, this 3-DSS is to give an assurance of quick update of data for the real-time information support. As a result, DSS of the 3-DSS is fragmented with a partitioning procedure. Therefore, there must have a coordination and communication among the tasks of the functionalities of the 3-DSS for doing the simultaneous operation, interchanging of the tasks of the functionalities and the partitioning of the DSS. The regulator algorithm will play the role for making the coordination and communication among the tasks of the

functionalities with the help of some other algorithms. The regulator algorithm is given in Figure 7.

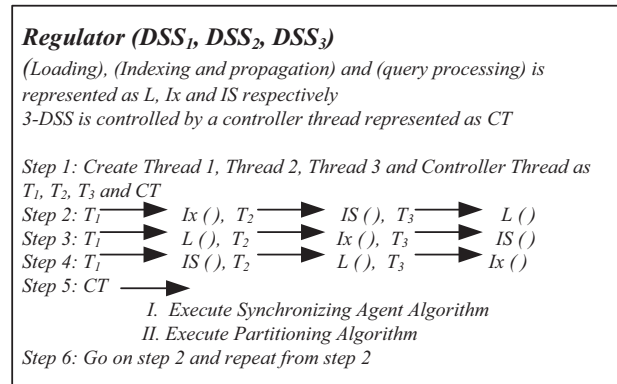


Figure 7. Regulator algorithm for 3-DSS

In the regulator algorithm of the 3-DSS, four threads will be created for the execution of the operation of 3-DSS. Thread 1, thread 2 and thread 3 are created for the simultaneous operation of the tasks of the functionalities of the 3-DSS in three individual DSS. On the other side, controller thread  $CT$  is constructed for the execution of the synchronizing agent and the partitioning algorithms. Synchronizing agent and partitioning algorithms are shown in Figure 8 and Figure 6 respectively. As thread 1, thread 2 and thread 3 work simultaneously, so, when thread 1 executes indexing function, thread 2 and thread 3 will execute the query and loading function, respectively. This will continue until thread 1, thread 2 and thread 3 get a message from the synchronizing agent of controller thread to change their tasks of the functionalities. If thread 1, thread 2 and thread 3 get a message from the synchronizing agent of controller thread to change their activities, then, thread 1 executes the function for loading task, thread 2 and thread 3 execute function for indexing and query processing tasks respectively. These activities of threading will continue until these threads do not get any message to interchange their activities. As soon as, each thread gets the message to interchange their activities, thread 1 will start query processing, thread 2 will start loading of data and thread 3 will start the indexing task. Hence, the sequence of activities of among threads will continue until the system is stopped by the user or any other reasons. For the shortening of indexing period, DSS will be partitioned by partitioning algorithm after a certain number of storage data. This partitioning process will provide the service from the controller thread together with synchronizing agent. Details of the synchronizing agent algorithm are given in Figure 8.



**Synchronizing Agent (DSS<sub>1</sub>, DSS<sub>2</sub>, DSS<sub>3</sub>)**

1.  $I_x()$ ,  $L()$  and  $IS()$  is executing simultaneously in  $T_1$ ,  $T_2$  and  $T_3$  by rotation.
2.  $I_x()$  And  $IS()$  inform CT of its completion status
3.  $L()$  sends a message to CT to get the status information of  $I_x()$  And  $IS()$
4. Waiting for the reply of CT about the status of  $I_x()$  And  $IS()$
5.  $IS()$  sends a message to CT after its completion to get the status information of  $I_x()$
6. Waiting for the reply of CT about the status of  $I_x()$
7.  $I_x()$  sends a message to CT after its completion to get the status information of  $IS()$
8. Waiting for the reply of CT about the status of  $IS()$
9. Completion of the status of  $I_x()$  And  $IS()$  = Boolean Value
10. If Boolean Value of  $I_x()$  = False AND Boolean Value of  $IS()$  = False
  - 10.1 Continue Current Functions in  $T_1$ ,  $T_2$  and  $T_3$
11. Else If Boolean Value of  $I_x()$  = True AND Boolean Value of  $IS()$  = False
  - 11.1 Continue Current Functions in  $T_1$ ,  $T_2$  and  $T_3$
12. Else If Boolean Value of  $I_x()$  = False AND Boolean Value of  $IS()$  = True
  - 12.1 Continue Current Functions in  $T_1$ ,  $T_2$  and  $T_3$
13. Else
  - 13.1  $T_1$  : Move to Next Function and Execute
  - 13.2  $T_2$  : Move to Next Function and Execute
  - 13.3  $T_3$  : Move to Next Function and Execute
14. Set
  - 14.1 Next Function  $\longrightarrow$  Current Function in  $T_1$
  - 14.2 Next Function  $\longrightarrow$  Current Function in  $T_2$
  - 14.3 Next Function  $\longrightarrow$  Current Function in  $T_3$

Figure 8. Synchronizing Agent algorithm for 3-DSS

Figure 8 presents the algorithm for the interchange process among of the tasks of the functionalities of the 3-DSS. It shows, how the tasks of the functionalities of the 3-DSS communicate with each other for providing their service rotationally in three individual DSS of the 3-DSS. According to the regulator algorithm of the 3-DSS, each task of the functionalities of the 3-DSS (indexing, loading and query processing) executes simultaneously in three separate threads by rotation. Further, each individual DSS of the 3-DSS changes role after a certain period of time. Query processing and indexing tasks are not possible to stop in the middle of the execution or before the completion of these tasks. Therefore, the indexing and the query processing tasks can be called dependent tasks. On the other hand, it is possible to stop the loading of data at any moment of time. Therefore, this task could be called independent task. Hence, the interchanging process of the tasks of the functionalities in the 3-DSS depends on both the indexing and the query processing tasks. In line 1 of algorithm indicates that step 1, step 2 and step 3 of regulator algorithm will be executed simultaneously by rotation. In line 2, function of the indexing and the query processing tasks will inform their current status to the controller thread. Then, the function of the loading task will send the message to the controller thread to know the current status of the indexing and the query processing function in line 3. The controller thread will deliver a reply about the status of the indexing and the query processing in line 4. In line 5 and 6, the query processing function will send a message to the controller thread to know the status of the indexing function after the completion its task and wait for the reply. Similarly, in line 7 and 8, indexing function will do the same and wait for the reply about the status of the query function. Now, from line 10 to line 13 shows that whether the tasks of the

functionalities of the 3-DSS will be interchanged or not. If the completion status of the query processing or the indexing function is false, the tasks of the functionalities of the 3-DSS will not be interchanged. So, from line 10 to line 12, current function is continued in thread 1, thread 2 and thread 3. In line 13, the completion status of both the query processing and the indexing function is true, so, the tasks of the functionalities of the 3-DSS is interchanged. For this reason, current function of each thread is stopped and move to the next function. Current function move to the next function in the regulator algorithm mean that indexing, loading and query processing function execute in step 1, step 2 and step 3 respectively in thread 1; query processing, indexing and loading function execute in step 1, step 2 and step 3 respectively in thread 2 and loading, query processing and indexing function execute in step 1, step 2 and step 3 respectively in thread 3. Now, if the current function of step 1 of thread 1, thread 2 and thread 3 are indexing, query processing and loading function respectively, next function will be the function of thread 1, thread 2 and thread 3 of step 2 and so on for step 2 and step 3. Therefore, when the next function will be prepared for execution, it will be executed as current function. Finally, in line 14, next function is set and executed as the current function in thread 1, thread 2 and thread 3. The whole process executes repeatedly to continue the interchanging of the tasks of the functionalities in three individual DSS simultaneously in the 3-DSS.

## V. ADDITIONAL TASKS FOR HANDLING THE SYSTEM FOR THE FAILURE OF PRINCIPAL 3-DSS

Two 3-DSS can be installed in IMS for the real-time information support seamlessly. One 3-DSS can be said principal 3-DSS. Another can be told alternative 3-DSS. The principal 3-DSS can be down at any moment of time in the system for the crashing or other difficulties for any of the DSS of the principal 3-DSS. An alternative 3-DSS can facilitate the seamless real-time time information support at the down period of the principal 3-DSS. Therefore, some of the additional tasks have to include in the regulator algorithm of Figure 7 for handling the system for the failure of the principal 3-DSS. These tasks will be executed in the controller thread. The additional tasks that will be included in the controller thread are,

**Sending the Source Data to Alternative 3-DSS:** The source data will be stored in the alternative 3-DSS at the same time of loading data in the principal 3-DSS. It is done by sending source data to one DSS of the alternative 3-DSS. This DSS then replicates the data to other two DSS of the alternative 3-DSS.

**Recovery System:** The log based or the shadow paging recovery system described in [19] can be used for recovering the data for crashing of 3-DSS.



**Activation of Alternative 3-DSS:** Alternative 3-DSS will be activated for information support just after the failure of principal 3-DSS. This alternative 3-DSS will then work just like the principal 3-DSS.

**Storage of Data in Temporary DSS:** The source data will also be stored in the temporary DSS for supporting the principal 3-DSS. It will store data as long as principal 3-DSS will be down.

**Transferring the Temporary DSS Data to Principal 3-DSS:** After fixing the problem of the principal 3-DSS, the stored data in the temporary DSS will now be transferred to the principal 3-DSS.

**Restart the Principal 3-DSS:** Now, the principal 3-DSS will be restarted and the activities of principal 3-DSS will be released from the alternative 3-DSS. This alternative 3-DSS will then perform its general task.

Now, the controller thread of the regulator algorithm in Figure 7 can be written as:

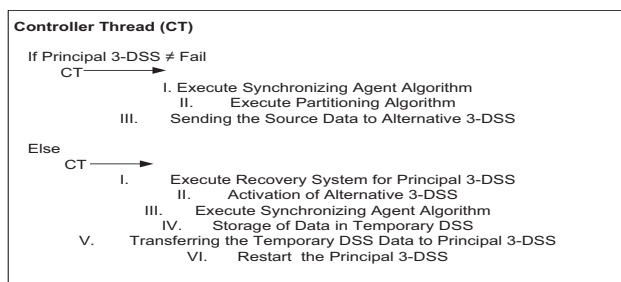


Figure 9. Additional tasks in controller thread for handling both principal and alternative 3-DSS

In Figure 9, if principal 3-DSS is not in failing state, then, the controller thread will execute the general 3-DSS tasks and Sending the source data to alternative 3-DSS task. On the other hand, if the principal DSS is in failing state, then, the controller thread will execute recovery system for principal 3-DSS. Then, it will activate the alternative 3-DSS. At the same time, synchronizing agent algorithm will start to work on the alternative 3-DSS. Storage of data in temporary DSS, transferring the temporary DSS data to principal 3-DSS and restarting the principal 3-DSS tasks will also execute in the controller thread. Once the principal 3-DSS restarts, the controller thread will again execute the tasks of not failing state.

## VI. EXPERIMENTAL EVALUATION

The experiments have been done for the 2-DSS and 3-DSS oriented IMS. 2-DSS is used as the benchmark for showing the real-time information support of 3-DSS. Temporary DSS is considered for the experiment of 2-DSS. Only loading of data task work in the temporary DSS of the

2-DSS for providing the real-time information support. The tasks of the DSS functionalities work in three individual DSS simultaneously in the 3-DSS. Four machines were used for doing the experiments. Among the four machines, three machines were used for implementing the 3-DSS, another is used as a server for controlling the 3-DSS, inserting data from the sources to the 3-DSS and sending the query request to the 3-DSS. Server machine and one more machine were used for the experiment of the 2-DSS. Multi core 2.2 Ghz processors, 4GB RAM and 5400 r.p.m hard drive were used for the server machine. The rest of the machine used single core 1.69 Ghz processor, 1GB RAM and 5400 r.p.m hard drive. SQL server was the database software for creating the data storage system.

For doing three experiments, 1GB data was stored in both the 2-DSS and the 3-DSS oriented IMS. Fifty thousand new rows (tuple) were extracted from the sources and stored in the 2-DSS and the 3-DSS with the refreshment function at the time of each individual experiment for delivering the data for the query request. One experiment of 3-DSS was done without storing 1GB data in the 3-DSS. In the real world, user may send the query request in the refreshment period. For this reason, query and refreshment functions executed simultaneously in these experiments. Query request for retrieving all newly inserted data was sent repeatedly after each single minute. Therefore, the query result was delivered for each respective query request. These query results were measured to get the result for both the 2-DSS and the 3-DSS oriented IMS. Start of data insertion time means the first data insertion time from the source to the DSS and query delivery time is the time the query result is delivered. Therefore, the distance between start of data insertion time and query delivery time is measured by subtracting query delivery time from the start of data insertion time. Volume of query result data is measured by dividing the number of inserted data in the DSS with the total data for insertion. The results are given in Table I, Table II, Table III, and Table IV.

TABLE I. EXPERIMENTAL RESULT OF 2-DSS (BENCHMARK DSS)

Query Request No.	Distance between Start of Data Insertion Time and Query Delivery Time (Minute)	Volume of Query Result Data (%)
1	1	11.00
2	2	20.80
3	3	31.20
4	4	41.27
5	5	50.30
6	6	61.16



TABLE II. EXPERIMENTAL RESULT OF 3-DSS (NO EXISTING DATA)

Query Request No.	Distance between Start of Data Insertion Time and Query Delivery Time (Minute)	Volume of Query Result Data (%)
1	1	10.70
2	2	20.30
3	3	30.60
4	4	41.15
5	5	50.00
6	6	60.80

TABLE III. EXPERIMENTAL RESULT OF 3-DSS (1GB EXISTING DATA)

Query Request No.	Distance between Start of Data Insertion Time and Query Delivery Time (Minute)	Volume of Query Result Data (%)
1	1	10.30
2	2	18.20
3	3	26.37
4	4	33.32
5	5	41.60
6	6	49.78

TABLE IV. EXPERIMENTAL RESULT OF 3-DSS (PARTITIONING)

Query Request No.	Distance between Start of Data Insertion Time and Query Delivery Time (Minute)	Volume of Query Result Data (%)
1	1	10.90
2	2	20.40
3	3	30.38
4	4	41.00
5	5	50.10
6	6	61.10

Table I is representing the experimental result for the 2-DSS. Table II, Table III and Table IV are showing the experimental result for the non-partitioned 3-DSS, the non-partitioned 3-DSS with 1 GB existing data and the partitioned 3-DSS with 1 GB existing data respectively. Query result of Table I and Table II is almost the same. There is a big difference between the query result of Table I and Table III. Indexing of data was the cause for this difference. It was not visible in the Table II for the low volume of data (only newly data was inserted and no existing data were there). As, partition was done after 1GB of data, Table IV presents almost the same volume of query result for each query request like Table I. Therefore, it can be said that 3-DSS can provide real-time information support. Further, as 3-DSS does not need the data transfer from non-indexed DSS to indexed DSS, it can provide information support seamlessly.

## VII. CONCLUSION AND FUTURE WORK

This paper showed that the 3-DSS model can provide seamless real-time information support from the IMS. It is quite possible to down any of the DSSs of the 3-DSS at the period of execution in the real world. Therefore, there should have a redundant or replication system of 3-DSS to provide information support service in the down period. Additionally, a recovery system needs to develop for this 3-DSS for recovering the data for the failure of the system for crashing or some other reasons. Replication and recovery system are described briefly in this paper. Further, details work is needed on the dynamic partitioning system. Therefore, future work will be conducted on the replication system of the 3-DSS, the recovery system of the 3-DSS and the dynamic partitioning system of the 3-DSS in a broader aspect. Additionally, comparison of 3-DSS oriented IMS with the 2-DSS oriented IMS will also be the future research work.

## REFERENCES

- [1] M. Bouzeghoub, F. Fabret and M. Matulovic-Broqué, "Modeling Data Warehouse Refreshment Process as a Workflow Application", Proceedings of the International Workshop on Design and Management of Data Warehouses, 1999, pp. 6.1-6.12.
- [2] D.P. Ballou and H.L. Pazer, "Modeling data and process quality in multi-input, multi-output information systems", Management Science, vol. 31, 1985, pp. 150-162.
- [3] C. Batini and M. Scannapieco, "Data Quality: Concepts, Methodologies and Techniques", Publisher: Springer, Berlin, Germany, 2006.
- [4] L. Bellatreche, K. Karlapalem, M. Mohania and M. Schneider, "What can partitioning do for your data warehouses and data marts?", IEEE, 2000, pp. 437-445.
- [5] C. Cappiello, C. Francalanci and B. Pernici, "Data Quality and Multichannel Services", PhD Thesis. Politecnico di Milano, 2005.
- [6] C. Cappiello, C. Francalanci and B. Pernici, "Time-Related Factors of Data Quality in Multichannel Information Systems", Journal of Management Information Systems, vol. 20, 2003, pp. 71-92.
- [7] C. Cappiello, C. Francalanci and B. Pernici, "A Self-monitoring System to Satisfy Data Quality Requirements", Springer Verlag, vol. 3761, 2005, pp. 1535-1552.



- [8] C. Cappiello and M. Helfert, "Analyzing Data Quality Trade-Offs in Data-Redundant Systems", *Interdisciplinary Aspects of Information Systems Studies*, Physica-Verlag HD, 2008, pp. 199-205.
- [9] B.A. Forouzan, C. Coombs and S.C. Fegan, "Data Communications and Networking", Publisher: Tata McGraw-Hill, 2003.
- [10] J.H. Hanson and M.J. Willshire, "Modeling a Faster Data Warehouse", *IEEE*, 1997, pp. 260-265.
- [11] Y. Hu, S. Sundara and J. Srinivasan, "Supporting Time-Constrained SQL Queries in Oracle", *Proceedings of the 33rd international conference on Very large data bases*, 2007, pp. 1207-1218.
- [12] W.H. Inmon, R.H. Terdeman, J. Norris-Montanari and D. Meers, "Data Warehousing for E-Business", J. Wiley & Sons, 2001.
- [13] M.V. Mannino and Z. Walter, "A framework for data warehouse refresh policies". *Decision Support Systems*, vol. 42, 2006, pp. 121-143 .
- [14] C.L. Pape and S. Gancarski, "Replica Refresh Strategies in a Database Cluster", *Springer-Verlag, LNCS* vol. 4395, 2007, pp. 679-691.
- [15] C.L. Pape, S. Gancarski and P. Valduriez, "Refresco: Improving Query Performance Through Freshness Control in a Database Cluster", *Springer-Verlag*, vol. 3290, 2004, pp. 174-193.
- [16] T. Padron-McCarthy and T. Risch, "Performance-Polymorphic Execution of Real-Time Queries. Workshop on Real-Time Databases: Issues and Applications", Newport Beach, California, USA, 1996. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.2149>. [Accessed 2<sup>nd</sup> January 2013].
- [17] J.F. Roddick and M. Schrefl, "Towards an Accommodation of Delay in Temporal Active Databases", *11th Australasian Database Conference (ADC)*, 2000.
- [18] R.J. Santos and J. Bernardino, "Real-Time Data Warehouse Loading Methodology", *ACM*, vol. 299, 2008, pp. 49-58.
- [19] A. Silberschatz, H.F. Korth and S. Sudarshan, "Database System Concepts". Publisher: McGraw-Hill, 1997.
- [20] D. Theodoratus and M. Bouzeghoub, "Data Currency Quality Factors in Data Warehouse Design", *Int. Workshop on Design and Management of Data Warehouses (DMDW)*, 1999.
- [21] A. Vavouras, S. Gatzia and K.R. Dittrich, "Modeling and Executing the Data Warehouse Refreshment Process", *IEEE*, 2000, pp. 66-73.
- [22] S.V. Vrbsky, "A data model for approximate query processing of real-time databases", *ACM Data & Knowledge Engineering*, vol. 21, 1996, pp. 79-102.
- [23] R.Y. Wang, M. Ziad and Y.W. Lee, "Data Quality", Publisher: Kluwer Academic, 2001.
- [24] T. Zurek and K. Kreplin, "SAP Business Information Warehouse From Data Warehousing to an E-Business Platform", *17th Int. Conf. on Data Engineering (ICDE)*, 2001.
- [25] K. Zdenek, M. Kamil, M. Petr and S. Olga, "On updating the data warehouse from multiple data sources", *Springer*, vol. 1460, 1998, pp. 767-775.



# Data Quality Comparison between Highly Integrated Single and Three Data Storage System Oriented Information Manufacturing System

Mohammad Shamsul Islam  
Faculty of Engineering & Computing  
Dublin City University (DCU)  
Dublin, Ireland  
Email: mohammad.islam6@dcu.ie

Markus Helfert  
School of Computing  
Dublin City University (DCU)  
Dublin, Ireland  
Email: markus.helfert@computing.dcu.ie

**Abstract**—Sources of the information manufacturing system can be integrated with the data storage system in different degrees. Lowly integrated information manufacturing system uses multiple data storage system for storing data in the information manufacturing system. As a result, lowly integrated information manufacturing system was the cause of some data quality problems. Therefore, highly integrated information manufacturing system is installed in the organizations. Highly integrated information manufacturing system store data in one single data storage system. Refreshment and query function are to execute in the data storage system of information manufacturing system for making data available and for information support respectively. The tasks of the refreshment function (data loading, indexing) execute sequentially in the single data storage system oriented information manufacturing system. Further, a highly integrated information manufacturing system is developed for executing the tasks of refreshment and query function simultaneously in data storage system. This highly integrated information manufacturing system is the 3- data storage system oriented information manufacturing system. These highly integrated information manufacturing system could also cause the data quality problems regarding the completeness, accuracy and timeliness data quality dimensions. Therefore, the purpose of this paper is to show a comparative data quality scenario of both of these highly integrated information manufacturing system. As a result, simulated highly integrated single and 3-data storage system oriented information manufacturing system is developed for assessing the data quality of these two information manufacturing system.

**Keywords**—accuracy; completeness; timeliness; single data storage system; 3-data storage system.

## I. INTRODUCTION

Multi channel or multi source information system, cooperative information system and web information system work as the information manufacturing system of an organization, inter-organizations and virtual organizations respectively. These information manufacturing systems work in both real time and non-real time environment. Data come from multiple channels or sources are to integrate to store data in the data storage system (DSS) of IMS of those

organizations. These data have to be processed before storing the data in the DSS of IMS. Processing of data is done with the refreshment function of the system. Until refreshment function is finished, data will not be stored for being available in the IMS. Therefore, longer the duration of refreshment period, there is a possibility of poor quality data [4,6].

Lowly integrated DSS oriented IMS can cause for data quality problems in IMS. Therefore, a highly integrated DSS oriented IMS is used to solve the data quality problems in IMS [5]. This highly integrated DSS oriented IMS store data in single DSS and deliver a set of data for information support for query request. Henceforth, refreshment and query function are to execute in single DSS oriented IMS for making data available and for the information support respectively. Furthermore, the tasks of the refreshment function (data loading, indexing) execute sequentially in the single DSS oriented IMS. Therefore, duration of refreshment process can be longer. According to [4], frequency of refreshment is high, timeliness of data is high. On the other hand, frequency of refreshment is low, availability of data is low. Furthermore, there is a possibility of approximate or incomplete data delivery from the system if, availability is low [19]. It is discussed in [5] that there is a time related accuracy and completeness problems for the refresh period in IMS. Therefore, data quality problems may occur in IMS for the frequency of refreshment and the duration of the refreshment process for the down time of the system and the obsolescence of data. Hence, if, it is possible to execute the tasks of the refreshment (data loading, indexing, and propagation of data) and query (query processing) function simultaneously, duration of refreshment could be shorter and refreshment could be continuous or frequent without affecting the availability and timeliness (obsolescence) of data. Highly integrated 3-DSS oriented IMS is needed for the simultaneous execution of the task of refreshment and query function. Henceforth, the purpose of this paper is to show a comparison of data quality between highly integrated single and 3-DSS oriented information manufacturing system (IMS).



## II. RELATED RESEARCH

Multi-channel information system is integrated to resist the redundant data in the operational data storage system in [5]. Multi channel integration is effective for the quality of stable or long-term changing data. Conversely, this integration is not suitable for frequently changing or time related data. In the IMS, it is difficult to maintain the quality of data for both timeliness and other objective data quality dimensions. Time related data integration for data warehouses are found in [20]. Continuous data integration is one of the important requirements for time related data storage system are discussed in this paper. According to Capiello and Helfert [4] analyzed the trade-off between availability and timeliness data quality dimension for synchronizing or refreshment frequency of DSS in IMS. Batini and Scannapieco [3] described the environment for the trade-off between the dimensions and the occurrence of trade-off between data quality dimensions. Basically, trade-off is done between the time-related data quality dimensions and objective data quality dimensions like completeness, accuracy and consistency. Theodoratos and Bouzeghoub [13] worked on the currency quality factors for data warehouse DSS. Data currency quality goal is expressed by currency constraint associated with every source relation in the definition of every input query. The upper bound in a currency constraint is set by the knowledge workers according to their needs. Mannino and Walter [10] showed in a survey that some organizations refresh data continuously, some organizations refresh data in every 5 minutes for updating data. Most organizations indicated that more frequent refresh during business hours would negatively impact on the system availability and the timeliness in IMS. The meaning of data warehouse updating is defined in [18]. According to this paper, data warehouse updating means a periodical data gathering, transformation and its addition into a data warehouse. Chaudhuri and Dayal [7] worked on the overview of the data warehouse DSS. According to them, Data warehouse DSS store the historical data or individual data record or consolidated data. This data is used for decision support. Considering the complexity of all the factors that are involved in the process of updating the data warehouse, Hanson and Willshire [9] model a faster data warehouse to make available the current data as soon as possible.

## III. HIGHLY INTEGRATED INFORMATION MANUFACTURING SYSTEM (IMS)

The information manufacturing system (IMS) is the information system that manufactures information from raw data [15]. The most important component of IMS is the data storage system (DSS). It is integrated with the multiple sources of the system. Therefore, it contains

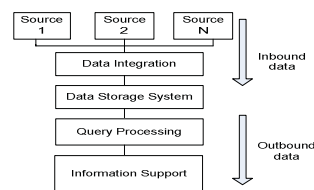


Figure 1. Information Manufacturing System

inbound raw data come from multiple sources. Data come from multiple sources are to be processed for the availability of data in the DSS of IMS. Available data in the DSS are then delivered by the processing of a query request as outbound data for information support.

According to [10] [11] [21], refreshment and query function are to execute in the data storage system of IMS to make the data available and for the information support respectively.

**Refreshment Function:** It is a complex process comprising the tasks, such as data loading, indexing and propagation of data for synchronizing data in the information manufacturing system (IMS) [10] [11] [21].

**Data loading:** Storage of manipulating [insert, update] data are to extract from the sources; transformed data if the source data are in the different format. After that, extracted and transformed data are to integrate and to clean for loading data in the data storage system [10] [11] [21].

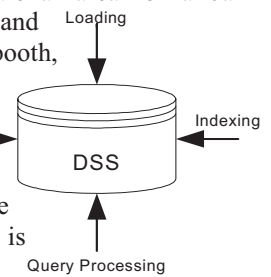
**Indexing:** Index is to update for newly loaded manipulated data or deleted data to align the data in the data storage system [11]. Indexing determines the effective usability of data collected and aggregated from the sources and increases the performance of the data storage system for information support [10] [11] [21].

**Propagation of Data:** Data is to propagate through the refreshment process for synchronizing the data of multiple DSSs of the system.

**Query Function:** This function of data storage system in IMS is done by the query processing task. The requested query of the user is processed in the data storage system for delivering the information to the user.

### A. Highly Integrated Single DSS Oriented IMS

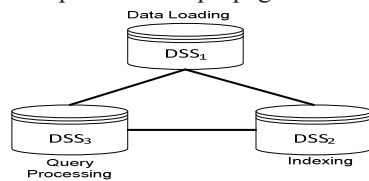
Single DSS is discussed in [6]. Data come from multiple sources are stored in highly integrated single DSS. Sources could be multiple functional areas or the multiple channels. Channels and functional areas are not same for all organizations. For example, functional area for a bank is trading, insurance, credit card etc. and channel could be internet, ATM booth, branches etc. Tasks of the query and refreshment function of data storage system run in single DSS. Therefore, execution process of the tasks of the refreshment and query function is sequential. It means that loading, indexing and query processing tasks work one after another and not simultaneously. Propagation of the data task does not need for the single DSS oriented IMS.





### B. Highly Integrated 3-DSS Oriented IMS

In the 3-DSS, three individual DSS are mutually interconnected with each other. The tasks of the refreshment and the query function of data storage system work simultaneously in three individual DSS. Therefore, data loading and indexing with updated data propagation task of the refreshment function work in two individual DSS of 3-DSS and another DSS of 3-DSS executes the task of the query function at the same time. After each successive period, the tasks of the refreshment and the query function of data storage system will interchange with cyclic order. As, propagation of manipulated data in the DSS is done simultaneously at the working period of the task of the functionalities, there may not have propagation delay. Therefore, 3-DSS will hold exactly the same data.



### Execution Process of 3-DSS

Suppose, DB1, DB2 and DB3 is three individual DSS for 3-DSS. These three DSS are mutually interconnected with each other. Now, if DB1 store some data from operational data sources, DB2 and DB3 must have the same data. The tasks of the query and refreshment function work simultaneously in these three data storage system. There must have a synchronization of starting and finishing time of the tasks of the query and refreshment function of these three data storage system. Manipulated data from operational data sources will load into one database. At the same time, another database will do the indexing and updated data propagation task for synchronizing data with other two DSS and the third one will be used for query processing. The indexed and query processing database lead the process. When the indexing with the propagation of updated data and the query processing are finished, the tasks of the refreshment and query function of data storage system will interchange with cyclic order. The rotation of the interchanging process is shown in the Figure 2.

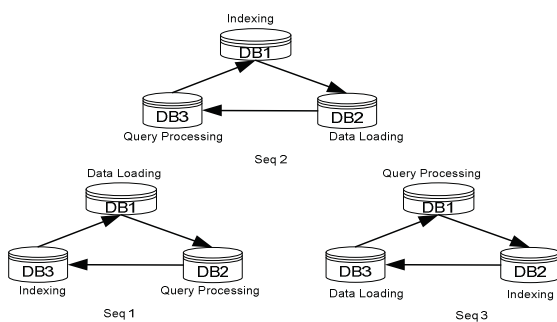


Figure 2. Rotation of the Tasks of Functionalities in 3-DSS

The algorithm for the rotation of function (Data Loading, Indexing and Query Processing) in data storage system is shown in Figure 3.

Rotation of Tasks of Functionalities in 3-DSS (Data Loading, Indexing with update propagation, Query Processing)	
DB1 = Database 1, DB2 = Database 2, DB3 = Database 3	
Step 1:	DB1⇒Data Loading, DB2⇒Nothing, DB3⇒Nothing
Step 2:	DB1⇒Indexing and send updated data to DB2, DB2⇒Data Loading, DB3⇒Nothing
Step 3:	DB2⇒Indexing and send updated data to DB3, DB3⇒Data Loading, DB1⇒Query Processing
Step 4:	DB3⇒Indexing and send updated data to DB1, DB1⇒Data Loading, DB2⇒Query Processing
Step 5:	DB1⇒Indexing and send updated data to DB2, DB2⇒Data Loading, DB3⇒Query Processing
Step 6:	Repeat Step 3, 4 and 5.

Figure 3. Algorithm for Rotation of Tasks of Functionalities in 3-DSS

In the algorithm, Steps 1 and 2 indicate the initialization of the system. Data come from multiple sources are loaded into DB1 in Step 1. DB1 executes the indexing task and send the updated data to DB2 and DB2 loads the updated data and source data simultaneously in Step 2. Each task of the functionalities of data storage system works simultaneously in Steps 3, 4 and 5. As, data have been loaded into DB2 in Step 2, DB2 is indexed in Step 3 and send the updated data to DB3. At the same time, DB3 loads the manipulated data including the updated data of DB2. Further, DB1 provides the information by processing the query request of the system in Step 3. Steps 4 and 5 will follow the same process but interchange the roles of each DB of the system. Therefore, Step 3, 4 and 5 will continue repeatedly in the system.

Details of the execution process of the 3-DSS are discussed in [22].

### IV. MOST USABLE DATA QUALITY DIMENSION IN IMS

According to researchers, most usable pertaining data quality dimensions are accuracy and completeness. On the other hand, time related data quality dimension is timeliness. The definition of these data quality dimensions is given below,

**Completeness:** Completeness of data in IMS is the ratio between the number of data stored in DSS and the number of data that should be stored in DSS [6]. Furthermore, if data is not available at the right time in IMS for the processing period, it cannot produce complete information [19].

**Accuracy:** Wang and Strong in [17] define the accuracy as the extent to which data are correct. Data stored in the DSS of IMS could be incorrect for the obsolescence. Further, according to [5], Data duplication or inconsistency as well as non-current data is caused for the inaccuracy of time related data.

**Timeliness:** Timeliness is defined as the extent to which data are timely for use [17]. It is also defined as the property of information to arrive early or at the right time [2]. Therefore,



timeliness of data in IMS depends on whether data are available in time or not. Obsolescence of data can be measured by the timeliness of data. Obsolete data are useless for the inaccuracy of data.

## V. EXPERIMENT FOR THE COMPARISON OF DATA QUALITY BETWEEN SINGLE DSS AND 3-DSS ORIENTED IMS

The following data quality assessment tool has been

developed for measuring the data quality for both single and 3-DSS oriented IMS. This tool is used for manipulating data in the IMS and delivery of the query request sending by the user in IMS simultaneously. In 3-DSS oriented IMS, tasks of the refreshment (loading, indexing + propagation of data) and the query function executed in three individual DSS of the IMS. Therefore, three individual DSS of 3-DSS have a communication with each other for changing their tasks from one DSS to another DSS. This complex 3-DSS oriented IMS will also controlled by this tool for measuring the data quality.

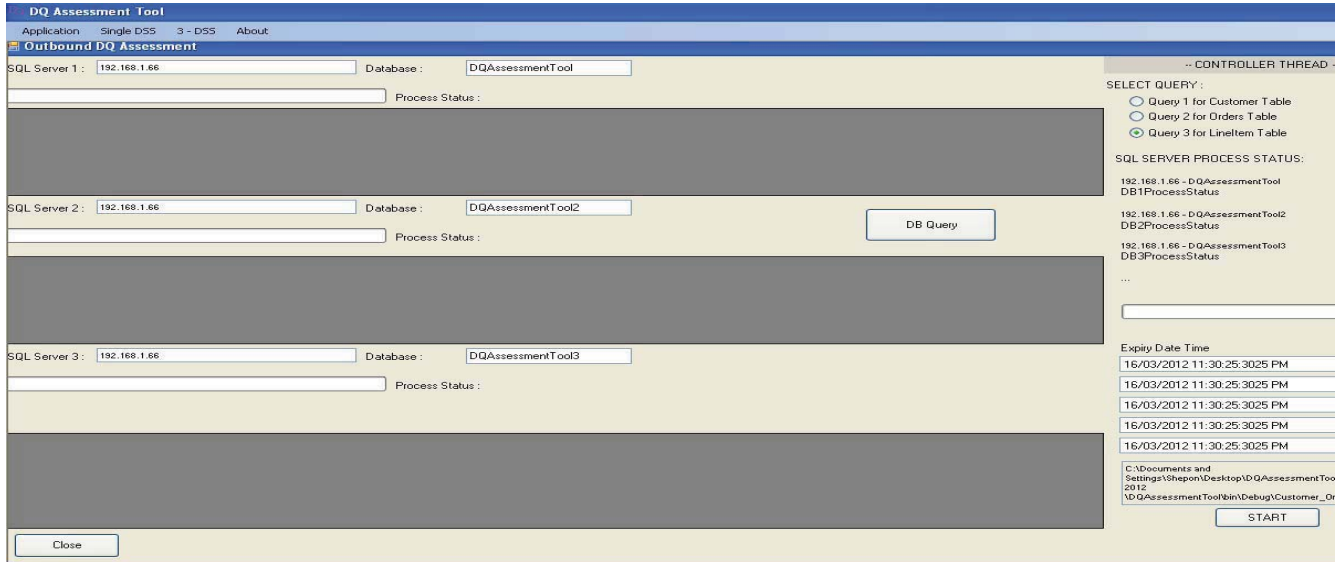


Figure 4. Data Quality Assessment Tool

### A. Assessment Functions for Measuring Data Quality in Highly Integrated IMS

Data quality is measured in the IMS by completeness, accuracy and timeliness. Therefore, these data quality assessment functions are implemented in this tool for measuring the data quality of both highly integrated single DSS and 3-DSS oriented IMS. The developed assessment functions for measuring completeness and accuracy of the information are taken from [23].

**Completeness:** From the completeness function of [5], following function can be written for measuring the completeness data in highly integrated IMS.

$$Completeness(C_k) = 1 - \frac{\sum_{i=1}^M \sum_{j=1}^N Incompleteness(d_{ij})}{M \times N} \dots\dots\dots (1)$$

$d_{ij}$  is the data for the particular location in DSS of IMS.  $i$  and  $j$  indicates the attribute (column) and tuple (row) respectively for measuring the completeness of data for a particular location of DSS.  $M$  and  $N$  are the total number of tuples (row) respectively. Therefore,  $M \times N$  is the total number of data in the DSS.

**Accuracy:** Considering the accuracy assessment function of [5], following accuracy assessment function can be written for measuring the accuracy of data in highly integrated IMS.

$$Accuracy(A_k) = 1 - \frac{\sum_{i=1}^M \sum_{j=1}^N Inaccuracy(d_{ij})}{M \times N} \dots\dots\dots (2)$$

Where,  $d_{ij}$  are the inaccurate elements of a particular location of IMS that does not contain the benchmark database or contain the benchmark database but obsolete for the time related issue or inaccurate for duplication issue.  $i, j$  indicates the attribute (column) and tuple (row) respectively for measuring the accuracy of the data for a particular location of DSS.  $M$  and  $N$  are the total number of attribute (column) and the total number of tuples (row) respectively. Therefore,  $M \times N$  is the total number of data in the DSS.

### B. Timeliness of Data in IMS

According to [3], Timeliness can be defined as currency and volatility dimensions. More specifically it can be written,

$$Max(0, 1 - currency/volatility) \dots\dots\dots (3)$$



This currency and volatility are to define for the DSS of IMS. The currency and volatility that are defined in [23,24] for the timeliness of data measurement is given in these papers.

**Volatility of Data in IMS:** As the definition of volatility we know that the length of time data remains valid is volatility [3]. Therefore, volatility of data depends on the expiry time of each individual data of DSS. Therefore, the formula for the volatility of data in the DSS of IMS is,

$$\text{Volatility} = \frac{\text{Expiry Time} - \text{Start of Data Insertion Time}}{\text{Expiry Time}} \dots\dots\dots (4)$$

Start of data insertion time means starting of insertion time of data from sources to the DSS of the information manufacturing system. Start of data insertion from the sources can be represented as SIT. On the other side, Expiry time can be represented as E<sub>T</sub>. Expiry time indicates the limit of the validity of data.

**Currency of Data in IMS:** According to [3], currency is defined as,

$$\text{Currency} = \text{Age} + (\text{Delivery Time} - \text{Input Time}) \dots\dots\dots (5)$$

Where Age measures how old the data unit is when received, Delivery Time is the time information product is delivered to the user and Input Time is the time data unit is obtained. Therefore, the currency dimension of data in the data storage system (DSS) depends on the age, delivery time and input time. In the database data storage system (DSS), these parameters can be recognized as below,

TABLE I. Currency Parameters of IMS

General Currency Parameter	DSS Currency Parameter	Notation
Age	Waiting Period + Refreshment Processing Period	W (t) + Rpro (t)
Delivery Time	Query Response Time	QRT
Input Time	Insertion Time of Data in DSS	IT

**Age A (t):** It can be calculated in DSS by adding waiting period of data with the refreshment processing period of data. Waiting period means how long data is waiting in the source before the refreshment processing of data in IMS for the insertion of data in the DSS. Refreshment processing period is calculated by adding the following parameters.

TABLE II. Refreshment Processing Period Parameters

Refreshment Processing Time Parameters	Description	Notation
Loading Period	Time needs for loading data in the DSS	L (t)
Indexing Period	Time needs for indexing data in the DSS	Ix (t)
Propagation Delay	Time needs for propagating data from one DSS to another DSS.	P (t)

Therefore, refreshment processing time of data in DSS can be calculated in the following way,

$$Rpro (t) = L (t) + Ix (t) + P (t) \dots\dots\dots (6)$$

Now, we can write the Age as,

$$A (t) = W (t) + (L (t) + Ix (t) + P (t)) \dots\dots\dots (7)$$

**Input Time:** To make data available in the DSS, data have to be inserted in the DSS of IMS. Data insertion will be completed if refreshment processing of data in the DSS is done. Therefore, the end of the refreshment processing time for each individual data will be the input time of individual data.

**Delivery Time:** It is defined in DSS by query response time. This query response time of DSS means, what time query request of a user query is responded in DSS.

### C. Execution Process of Experiment

This tool is used in a multi core 2.2 Ghz processors, 4 GB RAM, 5700 rpm hard disk based machine for doing the data quality experiment of both single and 3-DSS oriented IMS. SQL server was the database software for creating the data storage system. Same volume of data is stored in both types of DSS of IMS from the sources. Data come from the multiple sources of IMS was not 100% good quality. These data was 97% and 95% complete and accurate in the sources respectively. Data is extracted from the sources and stored in DSS with the refreshment function for delivering the data for the query request. In the real world, user can send the query request in the refreshment period. For this reason, query and refreshment function executed simultaneously in these experiments. A query result is delivered for each respective query request sending by the user. Therefore, these query results are measured by the data quality assessment function to get the data quality result for both single and 3-DSS oriented IMS. Timeliness value 0.3 is considered for measuring the obsolescence of data. if timeliness of data is greater than 0.3, data is accurate otherwise inaccurate.



TABLE III. Outbound Data Quality (DQ) Assessment Result for Single DSS Oriented IMS

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-240370	11:24:02	11:27:21	199437	633983	0.68	17.56	94.89
2	0-2385490	11:24:02	11:28:03	241989	633983	0.62	36.86	63.45
3	0-2931580	11:24:02	11:29:00	298972	633983	0.53	52.73	48.60
4	0-3312070	11:24:02	11:29:33	331872	633983	0.48	61.58	37.52
5	0-3904670	11:24:02	11:30:37	395069	633983	0.38	87.40	18.12

TABLE IV. Outbound Data Quality (DQ) Assessment Result for 3-DSS Oriented IMS

User	Age	Input Time	Delivery Time	Currency	Volatility	Timeliness	Completeness (%)	Accuracy (%)
1	0-130300	23:24:02	23:27:16	194535	633983	0.69	20.00	95.58
2	0-2263400	23:24:02	23:28:04	241914	633983	0.62	39.86	75.65
3	0-2901530	23:24:02	23:28:56	294972	633983	0.53	59.80	55.72
4	0-3312000	23:24:02	23:29:28	329671	633983	0.48	70.00	45.52
5	0-3862630	23:24:02	23:30:30	393069	633983	0.38	93.40	22.12

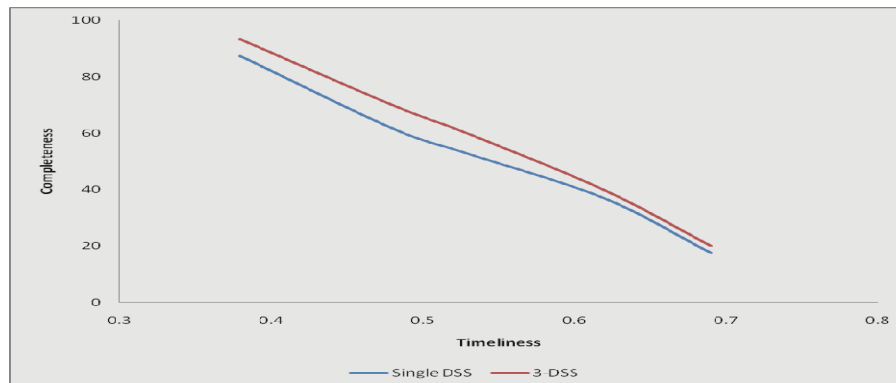


Figure 5. Outbound Data Quality (Completeness) Comparison between Single DSS and 3-DSS Oriented IMS



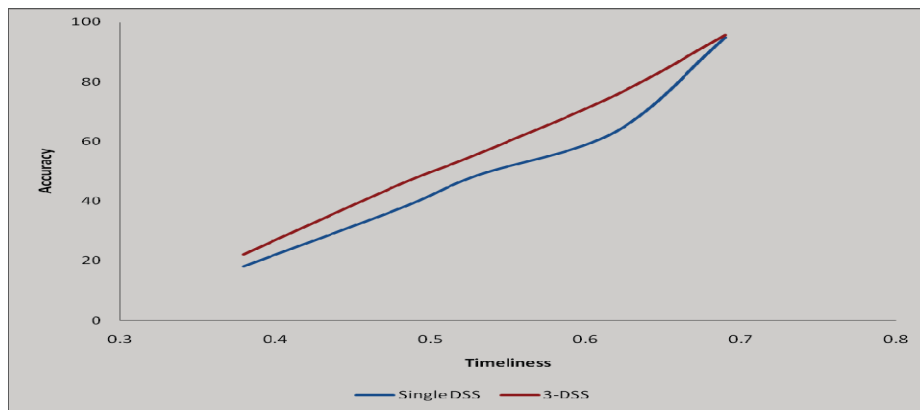


Figure 6. Outbound Data Quality (Accuracy) Comparison between Single DSS and 3-DSS Oriented IMS

#### D. Analysis

Table 3 and table 4 presents outbound data quality result for single and 3-DSS oriented IMS respectively. In these tables, 0 is the age of the first inserted data in DSS and input time is the time of the first inserted data. Therefore, timeliness of these tables is measured by the age and input time of the first inserted data and the delivery time of the set of data or information. Each individual data that is inserted in DSS has the age and input time. Delivery time varies in these tables. As a result timeliness of these tables varies for the delivery time. The set of data or information is coming at the right time or not depends on the delivery time. Completeness and accuracy comparison of single and 3-DSS oriented IMS are shown in the graph of figure 5 and figure 6 respectively. In the table 3 and table 4, it is found that volatility of data was indifferent for the experiment of both single and 3-DSS oriented IMS. Now, If we look at the distance between input time and the delivery time of each query request of both DSS oriented IMS, it will be seen that it is few seconds higher in single DSS oriented IMS than the 3-DSS oriented IMS for each query request. It means that query responding time for each query request of single DSS oriented IMS was few second late than the corresponding query request of 3-DSS oriented IMS. However, the data quality scenario of 3-DSS oriented IMS is better than the data quality scenario of single DSS oriented IMS.

The tasks of the refreshment and query function execute simultaneously using the multithreading process in 3-DSS oriented IMS. On the other hand, refreshment and query function execute sequentially in the single DSS oriented IMS for not defining the individual thread for each task. Therefore, the refreshment period of 3-DSS oriented IMS is shorter than the single DSS oriented IMS. For this reason, the age of the data of 3-DSS oriented IMS is lower than the age of the data of the single DSS oriented IMS. This age of the data affects on the accuracy data quality dimension with timeliness. Further, more data are inserted in 3-DSS oriented IMS than single DSS oriented IMS within the same duration of refreshment time.

Completeness of the data with timeliness varies for the insertion of data within the refreshment period. As a result, accuracy and completeness with timeliness of 3-DSS oriented IMS is better than the accuracy and completeness with timeliness of single DSS oriented IMS

#### VI. CONCLUSION

This paper compares the data quality between single and 3-DSS oriented IMS. By Comparing the data quality result of both single and 3-DSS oriented IMS, it is found that 3-DSS oriented IMS can provide better quality information with timeliness than the single DSS oriented IMS. But, there is no existence of 3-DSS oriented IMS in the real world organization. Therefore, future work will be conducted for the implementation of 3-DSS oriented IMS.

#### References

- [1] F. Arque "Real-time Data Warehousing with Temporal Requirements", *CEUR Workshop Proceedings*, vol. 75, 2003.
- [2] D.P. Ballou, R.Y. Wang, H.L. Pazer and G.K. Tayi, "Modeling Information Manufacturing System to Determine Information Product Quality", *Management Science*, vol. 44, 1998, pp. 462-484.
- [3] C. Batini and M. Scannapieco, "Data Quality: Concepts, Methodologies and Techniques", Publisher: Springer, Berlin, Germany, 2006.
- [4] C. Cappiello and M. Helfert, "Analyzing Data Quality Trade-Offs in Data-Redundant Systems", *Interdisciplinary Aspects of Information Systems Studies*, Physica-Verlag HD, 2008, pp. 199-205.
- [5] C. Cappiello, C. Francalanci and B. Pernici, "Time-Related Factors of Data Quality in Multichannel Information Systems", *Journal of Management Information Systems*, vol. 20, 2003, pp. 71-92.
- [6] C. Cappiello, C. Francalanci and B. Pernici, "A Self-monitoring System to Satisfy Data Quality Requirements", *Springer Verlag*, vol. 3761, 2005, pp. 1535-1552.
- [7] S. Chaudhuri, and U. Dayal, (1997) "An Overview of Data Warehousing and OLAP Technology", *ACM SIGMOD Record*, vol. 26, 1997, pp. 65-74.
- [8] C. Dong, M. Sampaio and F. Sampaio, "Expressing and Processing Timeliness Quality Aware Queries: The DQ<sup>2</sup>L Approach", *Springer Verlag*, vol. 4231, 2006, pp. 382-391.
- [9] J.H. Hanson and M.J. Willshire, "Modeling a Faster Data Warehouse", *IEEE*, 1997, pp. 260-265.
- [10] M.V. Mannino and Z. Walter, "A framework for data warehouse refresh policies". *Decision Support Systems*, vol. 42, 2006, pp. 121-143 .



- [11] R.J. Santos and J. Bernardino, "Real-Time Data Warehouse Loading Methodology", ACM, vol. 299, 2008, pp. 49-58.
- [12] M. Sampaio, C. Dong and F. Sampaio "Incorporating the Timeliness Quality Dimension in Internet Query Systems", Springer-Verlag, vol. 3807, 2005, pp. 53-62.
- [13] D. Theodoratus and M. Bouzeghoub, "Data Currency Quality Factors in Data Warehouse Design", Int. Workshop on Design and Management of Data Warehouses (DMDW), 1999.
- [14] P. Vassiliadis, M. Bouzeghoub and C. Quix, "Towards Quality-Oriented Data Warehouse Usage and Evolution", Journal of Information System, vol. 25, 2000, pp. 89-115.
- [15] R.Y. Wang, M. Ziad and Y.W. Lee, "Data Quality", Publisher: Kluwer Academic, 2001.
- [16] Y. Wand and R.Y. Wang, "Anchoring data quality dimensions in ontological foundations", Communications of the ACM, vol. 39, 1996, pp. 86-95.
- [17] R.Y. Wang and D.M. Strong, "Beyond accuracy: What data quality means to data consumers", Journal of Management Information Systems, vol. 12, 1996, pp. 5-33.
- [18] K. Zdenek, M. Kamil, M. Petr and S. Olga, "On updating the data warehouse from multiple data sources", Springer, vol. 1460, 1998, pp. 767-775.
- [19] S.V. Vrbsky, "A data model for approximate query processing of real-time databases", ACM Data & Knowledge Engineering, vol. 21, 1996, pp. 79-102.
- [20] R.M. Bruckner, B. List and J. Schiefer, "Striving towards Near Real-Time Data Integration for Data Warehouses", Springer-Verlag Berlin Heidelberg, vol. 2454, 2002, pp. 317-326.
- [21] M. Bouzeghoub, F. Fabret and M. Matulovic-Broqué, "Modeling Data Warehouse Refreshment Process as a Workflow Application", Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99), 1999, pp. 6.1-6.12.
- [22] M.S. Islam and P. Young, "Modeling a Data Storage System (DSS) for Seamless Real-Time Information Support from Information Manufacturing System (IMS)", The Fifth International Conference on Advances in Databases, Knowledge and Data Applications (DBKDA), 2013, pp. 134-142.
- [23] M.S. Islam, "An Assessment For Focusing The Change Of Data Quality (DQ) With Timeliness In Information Manufacturing System (IMS)", The Second International Conference on Digital Enterprise and Information Systems (DEIS), 2013, pp. 184-193.
- [24] M.S. Islam, "Regulators Of Timeliness Data Quality Dimension For Changing Data Quality In Information Manufacturing System (IMS)", The Third International Conference on Digital Information Processing and Communications (ICDIPC), 2013, pp. 126-133.