FRAMEWORK FOR INTEROPERABLE AND DISTRIBUTED EXTRACTION-TRANSFORMATION-LOADING (ETL) BASED ON SERVICE ORIENTED ARCHITECTURE

MOHAMMED M. I. AWAD

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2012

Awang Had Salleh Graduate School of Arts and Sciences
UUM CAS
Engaging Minds for a Better Tomorrow

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

MOHAMMED M. I. AWAD

calon untuk Ijazah PhD
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"FRAMEWORK FOR INTEROPERABLE AND DISTRIBUTED EXTRACTION-TRANSFORMATION-LOADING (ETL) BASED ON SERVICE ORIENTED ARCHITECTURE"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***19 Oktober 2011.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
October 19, 2011.

Pengerusi Viva: *(Chairman for VIVA)* — Prof. Dr. Norshuhada Shiratuddin — Tandatangan *(Signature)*

Pemeriksa Luar: *(External Examiner)* — Assoc. Prof. Dr. Wan Mohd Nasir Wan Kadir — Tandatangan *(Signature)*

Pemeriksa Dalam: *(Internal Examiner)* — Dr. Nor Laily Hashim — Tandatangan *(Signature)*

Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)* — Dr. Mohd Syazwan Abdullah — Tandatangan *(Signature)*

Tarikh:
(Date) *October 19, 2011*

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Pengekstrakan, Transformasi dan Proses Pemuatan (ETL) merupakan fungsi-fungsi utama dalam penyelesaian gudang data. Kekurangan komponen pengagihan dan saling kendali menyebabkan wujudnya lompang yang menimbulkan banyak masalah dalam domain ETL. Ini terjadi kerana komponen-komponen dalam kerangka kerja ETL sedia ada adalah saling berkait. Kajian ini membincangkan bagaimana untuk mengagihkan komponen-komponen ETL supaya komponen pengagihan dan saling kendali dapat dilaksanakan. Tambahan pula, kajian ini menunjukkan bagaimana kerangka kerja ETL dapat diperluaskan. Untuk mencapai tujuan tersebut, Perkhidmatan Berorientasikan Seni Bina digunakan untuk memperjelaskan ciri-ciri pengagihan dan saling kendali yang tidak wujud sebelum ini, dengan cara menyusun semula kerangka kerja ETL. Kajian ini menyumbang kepada bidang ETL dengan penambahan konsep pengagihan dan saling kendali kepada kerangka kerja ETL. Seterusnya, kajian ini juga menyumbang kepada bidang penggudangan data dan kepintaran perniagaan kerana ETL merupakan konsep utama dalam bidang ini. Metodologi Design Science Approach (DSA) dan Scrum digunakan untuk mencapai matlamat kajian ini. Integrasi di antara kedua-dua metodologi tersebut dapat mencapai objektif kajian ini. Kerangka kerja ETL yang baru ini direalisasikan menerusi pengujian dan penghasilan satu prototaip yang berdasarkan kepada kerangka kerja tersebut. Kejayaan prototaip ini dinilai berdasarkan tiga kajian kes yang melibatkan data dan alatan daripada tiga organisasi. Organisasi tersebut menggunakan penyelesaian gudang data untuk menghasilkan laporan statistik yang membolehkan pengurusan atasan membuat keputusan. Dapatan ketiga-tiga kajian kes ini menunjukkan komponen pergagihan dan saling kendali dapat dicapai dengan menggunakan kerangka kerja yang baru dalam ETL.

Katakunci: Pengekstrakan, Transformasi dan Proses Pemuatan, Gudang data, Sistem, Perkhidmatan berorientasikan seni bina.

Abstract

Extraction, Transformation and Loading (ETL) are the major functionalities in data warehouse (DW) solutions. Lack of component distribution and interoperability is a gap that leads to many problems in the ETL domain, which is due to tightly-coupled components in the current ETL framework. This research discusses how to distribute the Extraction, Transformation and Loading components so as to achieve distribution and interoperability of these ETL components. In addition, it shows how the ETL framework can be extended. To achieve that, Service Oriented Architecture (SOA) is adopted to address the mentioned missing features of distribution and interoperability by restructuring the current ETL framework. This research contributes towards the field of ETL by adding the distribution and interoperability concepts to the ETL framework. This leads to contributions towards the area of data warehousing and business intelligence, because ETL is a core concept in this area. The Design Science Approach (DSA) and Scrum methodologies were adopted for achieving the research goals. The integration of DSA and Scrum provides the suitable methods for achieving the research objectives. The new ETL framework is realized by developing and testing a prototype that is based on the new ETL framework. This prototype is successfully evaluated using three case studies that are conducted using the data and tools of three different organizations. These organizations use data warehouse solutions for the purpose of generating statistical reports that help their top management to take decisions. Results of the case studies show that distribution and interoperability can be achieved by using the new ETL framework.

Keywords: Extraction, Transformation and Loading, Data warehousing, Service Oriented Architecture.

List of Publications, Invited Speaker and Awards

Awad, M. M. I., & Abdullah, M. S. (2010). A Framework for Interoperable Distributed ETL Components Based on SOA. *Proceeding of the 2nd International Conference on Software Technology and Engineering (ICSTE 2010).*

Awad, M. M. I., & Abdullah, M. S. (2010). Extending ETL Framework using Service Oriented Architecture. *Procedia Computer Science Journal*, (3), 110-114.

Mohammed M I Awad - Keynote Speaker[1]. A Framework for Open-Source Interoperable Distributed ETL Components Based on SOA[2]. *Malaysia Open Source Conference 2010 (MOSC 2010).*

Best Award. *Malaysia Technology Expo 2011 (MTE 2011).*

Gold Medal. *Malaysia Technology Expo 2011 (MTE 2011).*

Silver Medal. *Seoul International Invention Fair 2011 (SIIF 2011),* South Korea.

Gold Medal. *Malaysia Technology Expo 2012 (MTE 2012).*

---

[1] http://conf.oss.my/speakers.html
[2] http://conf.oss.my/schedule.html

Acknowledgement

By the Name of Allah, the Most Gracious and the Most Merciful

My most profound thankfulness goes to my father and mother who have motivated me since my childhood to help me reach this success point. Thank you my parents and I hope to be the good son who can make you proud of him.

My deep thankfulness goes to my supervisor Dr. Mohd Syazwan Abdullah for supervising me during the journey of this research. I like his way of conducting the research, especially, his instructions for reviewing the literature, and the whole research as well.

Last but not least, I would like to thank my brothers and sisters as well as my friends.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

Acronym	Description
BI	Business Intelligence
DW	Data Warehouse
ETL	Extraction-Transformation-Loading
SOA	Service Oriented Architecture
J2EE	Java 2 Enterprise Edition
XML	eXtensible Markup Language
SOAP	Simple Object Access Protocol
DSA	Design Science Approach
WSDL	Web Services Description Language
LAN	Local Area Network
DSA	Data Staging Area
OMG	Object Management Group
CORBA	Common Object Request Broker Architecture
RMI	Remote Method Invocation
RPC	Remote Procedure Call
DCOM	Distributed Component Object Model
HTML	Hyper Text Markup Language
JSP	Java Server Pages
EJB	Enterprise Java Beans
DBMS	Database Management System
EIS	Enterprise Information System
HTTP	Hyper Text Transfer Protocol
JMS	Java Messaging Service
OLAP	online analytical processing
IT	Information Technology
OWB	Oracle Warehouse Builder
API	Application Programming Interface

BODI	Business Objects Data Integrator
CAL	client access license
SSIS	SQL Server Integration Services
GUI	Graphical User Interface
BIDS	Business Intelligence Development Studio
DCOM	Distributed Component Object Model
SC	Service Container
JDBC	Java Database Connectivity
BPEL	Business Process Execution Language

CHAPTER ONE
INTRODUCTION

This chapter presents the background and outlines the motivation of the research. This is followed by the research problems, the research question and the research objectives. Moreover, the research strategy is discussed and the scope of the research is argued. Furthermore, the research contribution is highlighted, the organization of the thesis is explored and chapter conclusions are presented.

1.1 Research Background

Data warehouses (DW) have become a main component of the corporate information system architecture, in which it plays a major role in building decision support systems (Vassiliadis *et al.*, 2002; Darmont *et al.*, 2005; Wrembel & Koncilia, 2007). By collecting data from a variety of internal and external sources, data warehouses use the transformation functionality which is a function in the ETL framework (explained in Chapter Two) to provide homogeneous information for analysis and reporting tasks (Wrembel & Koncilia, 2007; Bala *et al.*, 2009).

The uses of data warehousing products and services have been increasing over the years by industry as well as the development of the related technologies (Sen & Sinha, 2007; Wrembel & Koncilia, 2007). Furthermore, within the last decade, data warehouse field has made a vry important step by moving from simple centralized repositories to a platform for data integration and analysis (Vitt *et al.*, 2002; Mundy *et al.*, 2006; Watson & Wixom, 2007; Xi & Hongfeng, 2009). This move is pushing the success of the whole Business Intelligence (BI) field.

BI refers to techniques used in identifying, extracting and analyzing business data. These techniques provide historical, current and predictive views of business operations. Common functions of BI technologies are reporting, online analytical processing, analytics, data mining, business performance management, benchmarking, text mining and predictive analytics. These functions aim to support better business decision-making (Almeida *et al.*, 1999; Vitt *et al.*, 2002; Watson & Wixom, 2007; Tam, 2010). Furthermore, BI which is highly dependent on data warehousing; is successfully used together with warehouses in many industries including: healthcare, manufacturing, financial services, education, telecommunication, population, and other fields (Almeida *et al.*, 1999; Vitt *et al.*, 2002; Darmont & Boussaid, 2006; Mundy *et al.*, 2006; Tam, 2010).

Research problems related to creating, maintaining, and using data warehouse technology are somewhat similar to those specific for database systems. In other words, a data warehouse can be considered as a large database system with additional functionalities (Almeida *et al.*, 1999; Massachusetts, 2008; Silvers, 2008).

General database problems of index selection, materialized view maintenance, data integration, and query optimization have been reactivated in warehousing research (Vassiliadis *et al.*, 2002; Vassiliadis *et al.*, 2005; Tziovara *et al.*, 2007; Bâra *et al.*, 2008; Dessloch *et al.*, 2008; Siqueira *et al.*, 2009). On the other hand, some research problems are specific to data warehousing such as data acquisition, data cleaning, data warehouse refreshment, evolution of data warehouse schema, multidimensional and parallel query optimization, conceptual modeling for the data warehouses, data quality management, and data extraction, transformation and loading (ETL) enhancements (McCabe & Grossman, 1996; Bruckner *et al.*, 2002; Vassiliadis *et al.*, 2002; Du & Wong, 2004; Wehrle *et al.*, 2005; Zhang *et al.*, 2006; Sahama & Croll, 2007; Wehrle *et al.*, 2007; Santos & Bernardino, 2008; Mahboubi & Darmont, 2009).

ETL processes are meant to extract, transform and load the data into data warehouse for decision making (Wrembel & Koncilia, 2007). Effective ETL processes represent a major success factor for data warehouse projects and can absorb up to 80 percent of the time spent on any warehousing project (Vassiliadis *et al.*, 2002). These ETL processes are important because of the valuable functionalities that are performed using them. For example, they remove mistakes and correct missing data, provide documented measures of confidence in data, capture the flow of transactional data for safekeeping, adjust data from multiple sources to be used together, and structure data to be usable by end-user tools (Trujillo & Lujnmora, 2003; Kimball & Caserta, 2004; Tziovara *et al.*, 2007; Liao *et al.*, 2008).

Developing or using ETL is both a simple and complicated task. While the basic role of ETL is simply to get data out of the source and load it into the data warehouse (Sellis, 2006; Sen & Sinha, 2007; Morris *et al.*, 2008; Mrunalini *et al.*, 2009; Muñoz *et al.*, 2009; Simitsis *et al.*, 2009; Simitsis *et al.*, 2010), the development of ETL functionalities to incorporate additional requirements may break the ETL tasks into many little sub-cases, depending on data sources, business rules, existing software and destination reporting applications (Vitt *et al.*, 2002; Kimball & Caserta, 2004; Simitsis *et al.*, 2005). The development of these types of special requirements in current ETL tools is an uphill task to combine and reuse the broken little sub-cases and to keep perspective on the simple overall mission of the ETL system (Kimball & Caserta, 2004; Kshemkalyani & Singhal, 2008).

ETL processes perform at least three specific functionalities, and these are focused around the movement of data from one place or system to another (Sellis, 2006; Morris *et al.*, 2008; Mrunalini *et al.*, 2009; Shaikh *et al.*, 2010). These functionalities are: (1) the first function generally is to read data from an input source (file, relational table, or message queue); (2) to pass the stream of information through a process to modify, enhance, or eliminate data elements based on the instructions of the job; and (3) to take the resultant data and store it back to a file or relational table. These three steps are known as extraction, transformation and loading, respectively.

1.2 Research Motivation

The age of information technology (IT) is erasing the boundaries of cities, states, and countries. The success of IT applications depends on how remote distributed subsystems interoperate among each other's (Stonebraker & Hellerstein, 2001; Tanenbaum & Van Steen, 2002; Blair *et al.*, 2009; Li *et al.*, 2010). The biggest challenge is in aligning these independent subsystems into components that can interoperate across the enterprise branches through many locations (Coulouris *et al.*, 2001; Kshemkalyani & Singhal, 2008; Li *et al.*, 2010).

Data warehouses have occupied a big portion of those IT solutions to provide information for analytical processing, decision making, mining tools and other related technologies (Almeida *et al.*, 1999; Bruckner *et al.*, 2002; Du & Wong, 2004;

Inmon, 2005; Sahama & Croll, 2007; Santos & Bernardino, 2008; Mahboubi & Darmont, 2009).

Extract, Transform, and Load processes play a central role in the data warehouse solutions, and ETL is considered as the core component of a successful data warehouse system (Trujillo & Lujnmora, 2003; Muñoz *et al.*, 2009). ETL physically integrates data from multiple heterogeneous sources in a central repository referred to as data warehouse (Trujillo & Lujnmora, 2003; Kimball & Caserta, 2004; Muñoz *et al.*, 2009).

According to (Vassiliadis *et al.*, 2002), ETL consumes more than 60% of the data warehouse development effort. Furthermore, ETL and Data Cleaning tools are estimated to cost at least one third of the effort and expenses in the budget of the data warehouse projects, and this number can rise up to 80% of the development time in a data warehouse project. In addition, the ETL processes cost up to 55% of the total costs of data warehouse runtime.

Due to its importance and high cost, many research projects are carried out to enhance the ETL framework. Some of these projects in the last few years have concentrated on Real-Time data warehouse to solve the periodic Extraction, Transformation, and Loading problems (Bruckner *et al.*, 2002; Nelson & Wright, 2005; Abrahiem, 2007; Dou *et al.*, 2008; Santos & Bernardino, 2008). On the other hand, there is no sufficient research works that have been carried out to bridge the gap in this field regarding ETL components distribution and interoperability (Trujillo & Lujnmora, 2003; Kimball & Caserta, 2004; Simitsis *et al.*, 2005; Tziovara *et al.*, 2007; Wu *et al.*, 2007; Zhang & Wang, 2008; Skoutas *et al.*, 2009). That is because all of the previous research works view ETL as a tightly coupled software architecture rather than isolated components (Trujillo & Lujnmora, 2003; Kimball & Caserta, 2004; Simitsis *et al.*, 2005; Tziovara *et al.*, 2007; Zhang & Wang, 2008; Skoutas *et al.*, 2009), this is because the focus of these research works were on achieving other requirements of the ETL framework rather than involving distribution and interoperability features.

Data warehouse gets its data from many sources available in different separated locations (Kimball & Caserta, 2004). Each data source needs a complete ETL tool to sometimes do the extraction only (Henry *et al.*, 2005; Dung & Kameyama, 2007; Agrawal *et al.*, 2008; Mrunalini *et al.*, 2009; Suzumura *et al.*, 2010). Since an ETL tool is only available as one tightly coupled software (Kimball & Caserta, 2004; Apache, 2010; IBM, 2010; Microsoft, 2010; Oracle, 2010; SAS, 2010), there is no option other than installing all the ETL features together in each location, while the transformation and loading features are sometimes redundant for these sources. These two features would be necessary only in the location of the data warehouse destination and not the sources. Therefore, distributing ETL into loosely coupled and interoperable components can play a central role in solving this complication, and result in eliminating the redundant usage of many ETL licenses for the same project (Wu *et al.*, 2007). In addition, it can enable reusability, centralized ETL administration, ETL components portability, and ETL ease of use (Wu *et al.*, 2007).

Data warehouses normally include huge amounts of data that leads to slow report generation due to this massive amount of data (Almeida *et al.*, 1999; Vitt *et al.*, 2002; Tam, 2010). There are some hardware based solutions to this issue, but there is lack of research regarding extending the current ETL framework by an extra component to solve this complication; due to the tightly-coupled disadvantage of the current ETL framework (Vassiliadis *et al.*, 2002; Trujillo & Lujnmora, 2003; Kimball & Caserta, 2004; Vassiliadis *et al.*, 2005; Tziovara *et al.*, 2007; Zhang & Wang, 2008; Skoutas *et al.*, 2009).

1.3 Problem Statement

Data warehouses are complex systems employed to integrate the organization's data from several distributed and heterogeneous sources (Almeida *et al.*, 1999; Ault, 2003; Kimball & Caserta, 2004; Mundy *et al.*, 2006; Silvers, 2008). The heterogeneous sources are located in different locations far from each others. Each location has its own specific infrastructure for the systems running in that location. For example, it has its own operating system and deployment infrastructure such as .NET, J2EE, or IBM mainframe (Apache, 2010; IBM, 2010; Microsoft, 2010; Oracle, 2010; SAS, 2010). Furthermore, each of these infrastructures needs a special ETL tool to be compatible with. In addition, each data source needs a complete ETL tool to be installed in the same location of the source (Kimball & Caserta, 2004), while sometimes only the Extract function is needed to extract data from this source (Kimball & Caserta, 2004). This results in a problem of an increase in the ETL licenses needed for a DW project and an increase in the complexity for the ETL user due to the redundant processes (Wu *et al.*, 2007).

Sometimes, due to the complexity, long learning curve of the available ETL tools, and difficulty to achieve some extensibility in terms of additional functionalities; some organizations prefer to turn to in-house development to perform ETL tasks

(Kimball & Caserta, 2004; Inmon, 2005; Temenos, 2005), which increases the cost and effort of the data warehouse project (Wehrle *et al.*, 2007). Furthermore, tightly coupled components of a software require teams of architects and designers to untangle the complex implications of change in the support of new business requirements or system enhancements (Sneed, 2006; Wu *et al.*, 2007; Tam, 2010). As a result, tightly coupled components cause problems in terms of cost, maintenance, enhancement, and reusability of the ETL components (Wu *et al.*, 2007). This leads to the necessity that the architecture of the software framework to consider the component coupling factor and how much loosely the components should be coupled.

According to (Kimball & Caserta, 2004; Wrembel & Koncilia, 2007; Wu *et al.*, 2007), the current ETL framework lacks of components flexibility and loose coupling. This results in complications to add new components to the ETL tools; to support special business needs (Kimball & Caserta, 2004; Newcomer & Lomow, 2004; Kshemkalyani & Singhal, 2008). For instance, although data warehouses provide an appropriate infrastructure for efficient querying, reporting, mining, and other advanced analysis techniques (Inmon, 2005; Silvers, 2008), the complexity of data warehouse environments especially the ETL framework is rising every day, and data volumes are growing at a significant pace, which makes report generation relatively slow due to the massive amount of data (Inmon, 2005). According to (Wehrle *et al.*, 2007), data warehouse repositories based on one central server often suffer from either storage or computing bottlenecks, especially when complex aggregates need to be stored permanently or computed on demand. (Wehrle *et al.*,

2007) refers to a high cost solution to the massive data problem, which is cluster-type systems with large numbers of worker nodes connected through high-speed LAN.

In addition to the high expenses, integrating existing resources from several distant sites using this solution; needs a system that efficiently organizes these resources in a transparent manner, that at times is a challenge to implement (Pentaho, 2006; Pentaho, 2009). Some implementations of the ETL frameworks like "Pentaho Open Source Business Intelligence" include a fragmentation feature like Pentaho "Partitioning" (Pentaho, 2006; Pentaho, 2009). However, this feature does not classify data based on certain conditions to fulfill specific business needs. Furthermore, this fragmentation aims mainly to enable the fact and the dimension tables in the data warehouse to be separated among a cluster of servers. This belongs to a physical (hardware) solution of the performance problem, and such solution is outside the scope of this research. Therefore, an extensible ETL framework with loosely coupled components can resolve this complication, because, a specific component can easily be added as an extension to the framework to resolve the performance problem (Newcomer & Lomow, 2004; Wu *et al.*, 2007).

Administration of ETL tools in many data source locations for the same project to extract data from many different sources, requires additional administration, communication and maintenance effort (Wu *et al.*, 2007). This is a problem resulted from the reality that the administrators are often different persons from one location to another and they could use different ETL tools and do different configurations to these tools (Kimball & Caserta, 2004; Albrecht & Naumann, 2008). Furthermore, in the current ETL framework, there are impediments to include ETL as a part of a complete portal that manages the whole DW project because of complexity in the current ETL framework to communicate with other components of the portal (Wu *et al.*, 2007).

Therefore, there are gaps in the current ETL framework and these are related to: distribution and interoperability of the ETL components. These gaps lead to problems of the current ETL framework. These problems include: the complexity of extending the ETL tools to suit special business needs, the ETL administration complexity, and an increase of effort needed to implement a DW project. In addition, the distribution and interoperability gaps lead to: impediments regarding ETL compatibility with different administrator environments, an increase of the cost to implement a DW project due to the increase of the number of ETL licenses needed, and an extra effort needed to develop and use the ETL processes. Furthermore, the same gaps lead to a redundancy problem of including all the ETL features in every ETL administrator location due to the tightly coupled architecture of the available ETL framework.

As such, defining a conceptual framework for ETL that includes the features of components distribution and interoperability addresses the problems highlighted in this section.

1.4 Research Questions

As identified in section 1.3, there are problems due to the absence of components distribution and interoperability of the current ETL frameworks. Therefore, the main research question of this study is:

"Can a conceptual framework for ETL that includes the features of component distribution and interoperability be defined to enhance the current tightly coupled ETL framework?"

The main research question can be divided into three sub questions, which are:

i. What are the problems of the current ETL framework that result from the absence of components distribution and interoperability?
ii. How can the problems of the current ETL framework be tackled for defining distributed and interoperable components for ETL framework?
iii. How to define, test and evaluate the new ETL framework?

1.5 Research Objectives

The overall objective of this research is to define a conceptual framework for interoperable distributed ETL components. This framework is an enhancement to the current ETL framework in terms of components distribution and interoperability.

In particular, the research objectives are:

i. to identify problems of the current ETL framework due to the absence of component distribution and interoperability.

ii. to define components distributable and interoperable ETL conceptual framework.

iii. to demonstrate the applicability of the proposed ETL framework by the development of ETL prototype.

iv. to test and evaluate the distribution and interoperability of the prototype that validates the new ETL framework.

Each of the research objectives are achieved according to the research strategy (following the phases of the research methodology) in section 1.6.

1.6 Research Strategy

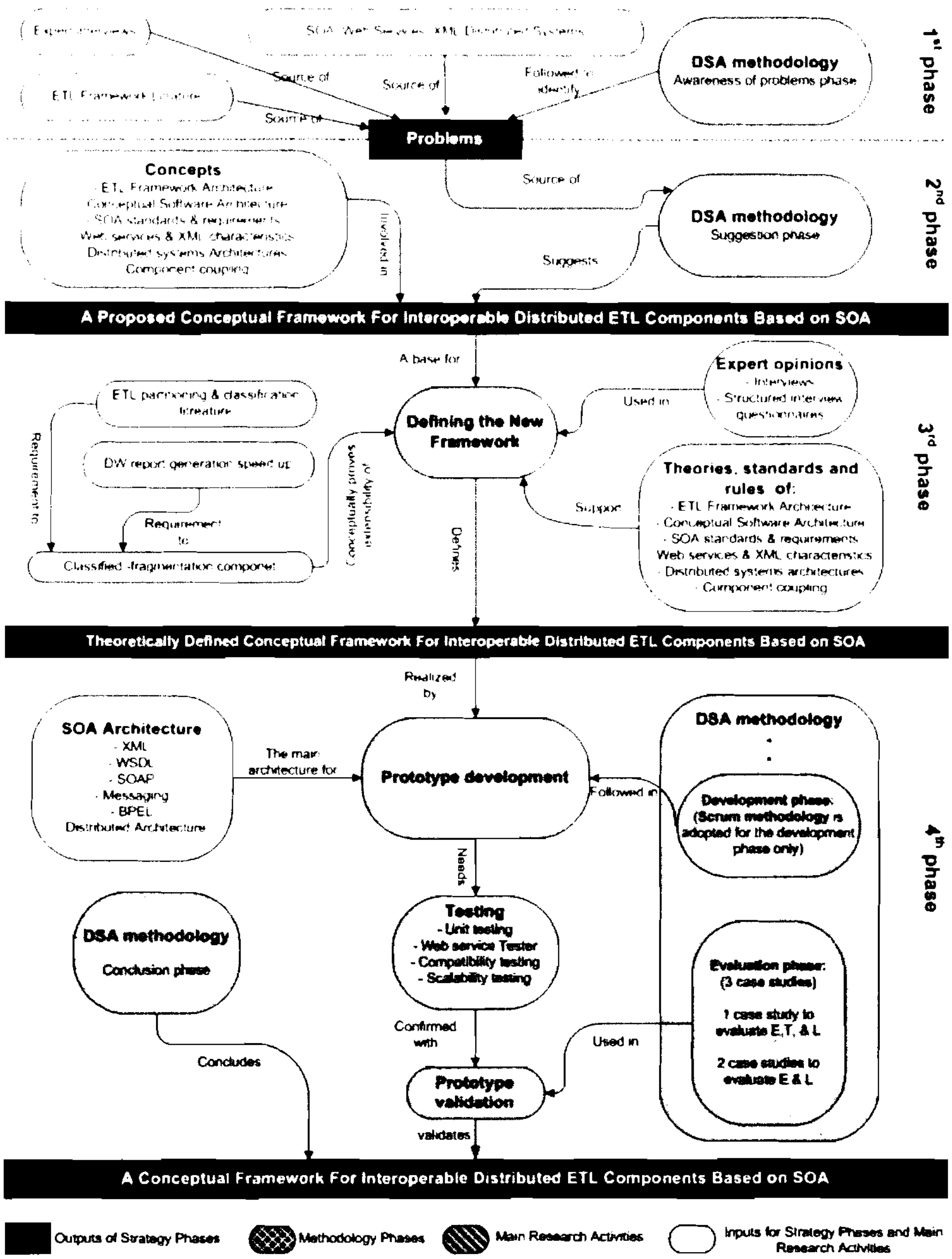

Figure 1.1: Research Strategy

The research strategy consists of four phases to achieve the research objectives and answer the research question. Figure 1.1 illustrates the research strategy. The 1st phase answer the question of: *What are the problems of the current ETL framework, which result from the absence of components distribution and interoperability?* The discussion of ETL literature, together with exploring the advantages of involving SOA, XML, distributed architecture and other related technologies are important to identify the available problems, which are resultant from the absence of components distribution and interoperability in the current ETL framework. Upon completion of this phase, a complete awareness of the problems is achieved.

The 2nd phase starts with analyzing the requirements of proposing the research work according to the suggestion phase of the DSA methodology. SOA framework plays a central role in resolving components distribution and interoperability issues of the newly defined framework. The research problems motivate for more research to bridge the gaps that have been left in the ETL field regarding distribution and interoperability. A literature about distributed systems, web services, SOA and software architecture is essential in determining the methods and standards that help in bridging the gaps. After determining the appropriate methods and following them in bridging the gaps; the resultant partial solutions are combined together to establish an initial conceptual framework for interoperable distributed ETL components. This framework is the output of the 2nd phase, and is also referred in the thesis as the "new ETL framework" or "restructured ETL framework".

The 3rd phase concentrates on defining the complete theoretical framework before realizing it by a prototype. Defining the framework depended on theories, standards and rules of ETL and SOA and other concepts related to SOA. In addition to that, expert opinions through structured interviews are utilized in defining the framework.

The 4th phase focuses on testing and validating the new framework. Specific methods that comply with the ETL and SOA frameworks are used to define, test and verify the framework. The framework is realized by developing a prototype for testing and validation purposes. Analysis, design, development, and deployment tools are used for the implementation of that prototype, and the prototype is evaluated using three case studies. In addition to evaluating the distribution and interoperability of the framework components, one case study is used to evaluate each of Extraction, Transformation and Loading components, while the other two case studies are used in evaluating Extraction and Loading components. The three case studies are explored in details in chapter 4 and chapter 7. Upon completion of this phase, the goal of the research is achieved and the conceptual framework has been defined, verified and validated.

1.7 Scope of the Research

This research concentrates on bridging the gaps of the current ETL Framework regarding components distribution and interoperability. Therefore, the scope of the research includes: ETL framework and processes; distribution concept which is limited to software techniques and does not use hardware techniques, as the hardware based techniques are already explored by previous researches (Wehrle *et al.*, 2007). Core architectures include 1-tier, 2-tier, 3-tier and N-tier; and applying distribution and interoperability techniques to the new framework is limited to SOA, web services, XML, and software distribution standards. The classification and fragmentation regarding types of data in data warehouse for the extended classified-fragmentation component is limited to text, image or/and video. All of these scopes are fully or partially used in the research. However, the focus of this research is generally to define a conceptual framework for interoperable distributed ETL components.

1.8 Contributions

The high-level goal of this research is to define a new ETL framework for Interoperable Distributed ETL Components based on Service Oriented Architecture. Therefore, this research contributes towards the field of ETL by adding the distribution and interoperability concepts to the ETL framework. Furthermore, this research adds contributions towards the area of data warehousing and business intelligence because ETL is a core concept in this area, particularly, in the area of data warehousing and business intelligence that covers the design, implementation and usage of ETL. This research:

- Contributes to the ETL field by involving the distribution concept among the ETL framework components, which enables the loose coupling among these ETL components. In addition, this research has provided interoperability in the interaction between clients (ETL administrators) and the loosely-coupled distributed ETL components by using SOA as the interaction architecture among the framework components.

- Contributes to the data warehouse and business intelligence area by providing distributable interoperable ETL framework to be involved in the design and implementation of data warehouse and business intelligence projects.

- Contributes to the ETL vendors, since it enables them to develop new releases of ETL tools including the distribution and interoperability features based on the new ETL framework. Furthermore, the new framework simplifies the process of extending and adding new components to the vendor's ETL tools, and reduces the cost and effort of developing ETL tools that are compatible with legacy systems. In addition, the new framework simplifies the reusability of ETL components since it enables ETL developers in DW industry to adopt the code of an existing ETL component developed by them or by any other ETL vendor who follows the new framework specifications, and then reuse it to meet new ETL business requirements. This reuse results in a huge savings in ETL tool development cost and time.

- Contributes to the ETL administrators (users) by simplifying the administration process of a DW project. The ETL administration can be centralized on one server because the ETL components can be deployed on one portal server. Therefore, the administrator focuses on a central unique ETL tool instead of administering many ETL tools at many geographical locations. Furthermore, the new framework contributes to the ETL administrators by eliminating the requirements and settings of the user machine (ETL administrator PC) such as operating system requirements and compatibility requirements, because the concepts of the new ETL framework allow ETL vendors to develop ETL components that can be deployed as parts of a business intelligence portal and the client can execute the ETL functionalities through the browser. However, ETL tools developed using the traditional ETL framework needs to be installed on the user machines (normally ETL administrator PCs), which requires special operating systems and special configurations and settings on the client machines.

- Contributes to the ETL customers (Companies that buy non-free ETL tools to implement DW projects) by reducing the number of licenses needed for implementing a DW project. Therefore, instead of having a license for every data source administrator, it is sufficient to have only one tool that can be deployed as components of a Business Intelligence portal and then every administrator can have an access to these components through his account on the portal.

1.9 Thesis Organization

The thesis consists of eight chapters; it starts with Chapter 1 that discusses the research background, motivation, problems, questions, objectives, strategy, scope and contribution. Then, Chapter 2 presents the literature of data warehouse, especially ETL framework, followed by discussion on the SOA framework which is very essential to restructure the ETL framework in Chapter 3. Chapter 4 discusses the research methodology that utilizes suitable methods for executing this research. This is followed by Chapter 5 that conceptually defines the theoretical framework. Discussion on the prototype is highlighted in Chapter 6, while Chapter 7 presents the evaluation of the prototype. Finally, Chapter 8 concludes the research and presents the future works.

1.10 Conclusion

This chapter has introduced the research background, presented the motivation of the research and highlighted the research problems, objectives, scope and contribution. Chapter 2 critically discusses the literature related to the traditional (current) ETL framework.

CHAPTER TWO

EXTRACTION, TRANSFORMATION, AND LOADING IN DATA WAREHOUSING

This chapter discusses about data warehousing, the ETL framework components and commercial ETL tools. It also highlights the ETL components coupling, distribution and compatibility. In addition, it argues the ETL extensibility, scalability, administration and licensing.

2.1 Data Warehouse

A data warehouse is a central repository for all or significant parts of the data that an enterprise's various business systems collect (Inmon, 2005) and it has become a main component of the corporate information system architecture (Almeida *et al.*, 1999; Holzer *et al.*, 1999; Massachusetts, 2008; Silvers, 2008). A data warehouse model can be classified into two parts, back room and front room. As shown in Figure 2.1, these are physically, logically, and administratively separated. In other words, the back and front rooms are on different machines. They follow different data structures, and are managed by different IT specialists (Vassiliadis *et al.*, 2002; Kimball & Caserta, 2004; Tziovara *et al.*, 2007). Data management for data warehouses involves acquiring data, transforming and delivering that data to the query-friendly front room. No query services are provided in the back room. Data access is prohibited in the back room, and therefore, the front room is dedicated only for this purpose (Kimball & Caserta, 2004; Santos & Bernardino, 2008; Zhu *et al.*, 2008a).

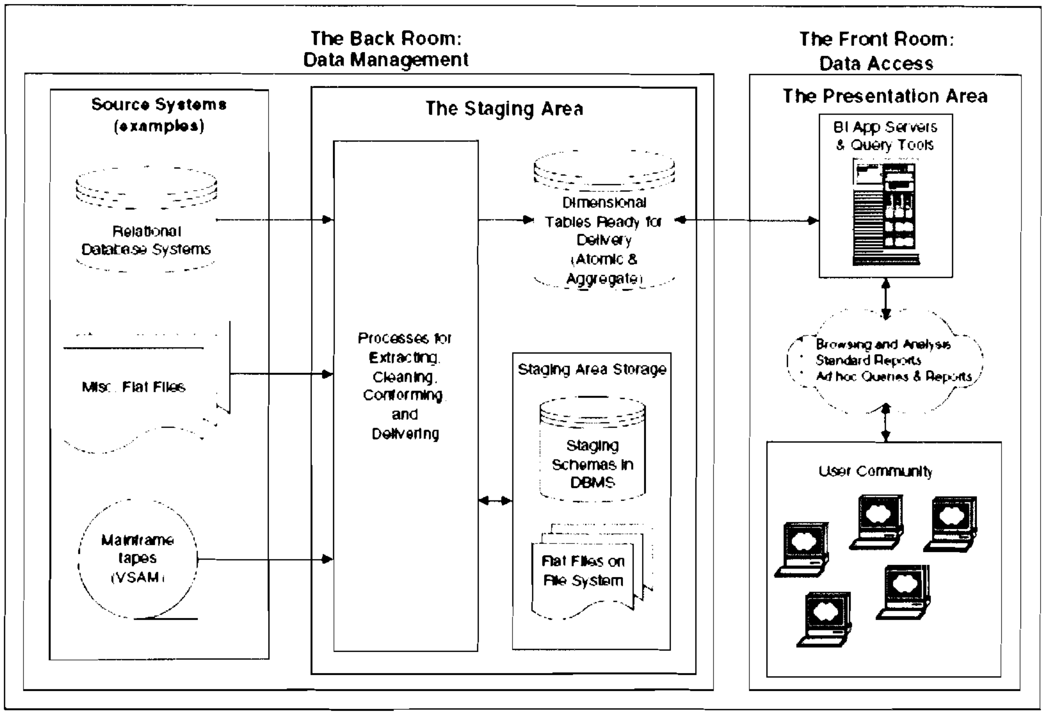

Figure 2.1: General Data Warehouse Components (Kimball & Caserta, 2004)

The staging area is a back-room facility, where the data is placed after it is extracted from the source systems, cleaned, manipulated, and prepared to be loaded to the presentation layer of the data warehouse. Any meta-data generated by the ETL processes that is useful to end users must be from the back room and is offered in the presentation area of the data warehouse (Kimball & Caserta, 2004; Microsoft, 2009; Oracle, 2009; Pentaho, 2009; SAS, 2010).

In this chapter, the literature regarding ETL research works to highlight the gaps of these works by showing where recent stages in this area have reached, and are critically discussed.

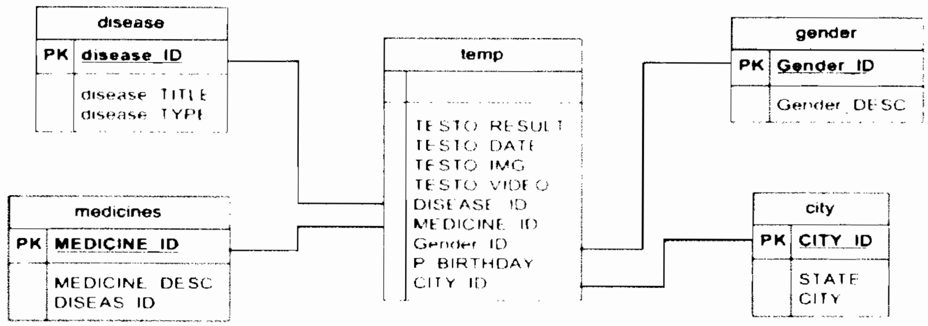

Figure 6.2: EXTRACT TEMP STORAGE Database

b. *TRANSFORM TEMP STORAGE database*

A TRANSFORM TEMP STORAGE database is a temporary storage that is generated as a result of the transformation process done on the EXTRACT TEMPSTORAGE database. The schema of this database is a star schema that consists of one fact table, and four dimension tables. The detailed explanation about every table and every field in this schema is described in Appendix A.

The transformation of data that is done as a sample of transforming data; is to transform P_BIRTHDAY to P_AGE.

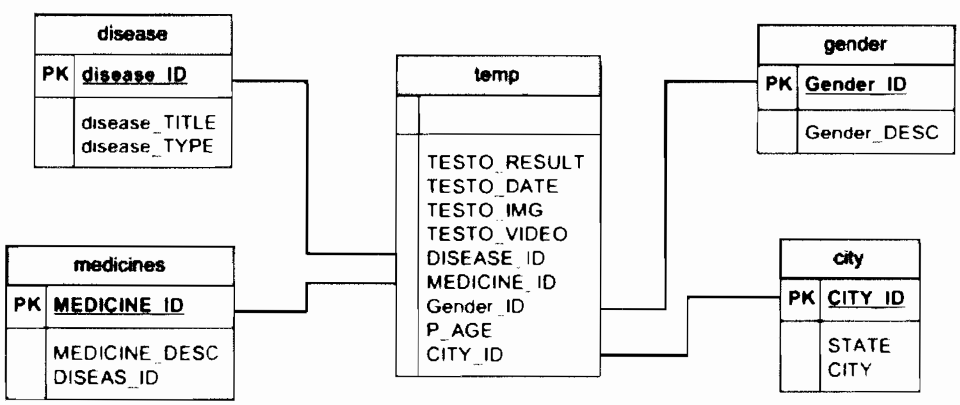

Figure 6.3: TRANSFORM TEMP STORAGE Database

```
        }
        return v;
    }

    public int getYear(String d) {
        int year;
        String[] split = d.split("-");

        year = Integer.parseInt(split[0]);
        return year;
    }
}
```

Table B.4: LoadData.java

```
package data.etl.dw;

import java.sql.*;
import java.util.*;
import java.io.*;

public class LoadData {

    private String host;
    private String port;
    private String user;
    private String password;
    private String db;
    private String url;
    private Connection conn;

    public LoadData(String config) throws IOException {
        Properties myproperties = new Properties();

        FileInputStream in = new FileInputStream(config);
        myproperties.load(in);
        in.close();

        this.host = myproperties.getProperty("host");
        this.port = myproperties.getProperty("port");
        this.user = myproperties.getProperty("user");
        this.password = myproperties.getProperty("password");
        this.db = myproperties.getProperty("db");
```