▩ 응용논문

데이터웨어하우스에서 이질적 형태를 가진 데이터의 추출을 위한 Extraction Transformation Transportation(ETT) 시스템 설계 및 구현

Extraction Transformation Transportation (ETT) System Design and implementation for extracting heterogeneous Data on Data Warehouse

여성주*
Sung-joo Yeo
왕지남*
Gi-nam Wang

Abstract

Data warehouse(DW) manages all information in a Enterprise and also offers the specific information to users. However, it might be difficult to develope an effective DW system due to the varieties in computing facilities, data base, and operating systems. The heterogeneous system environments make it harder to extract data and to provide proper information to users in real time. Also commonly occurred is data inconsistency of non-integrated legacy system, which requires an effective and efficient data extraction flow control as well as data cleansing. We design the integrated automatic ETT(Extraction Transformation Transportation) system to control data extraction flow and suggest implementation methodology. Detail analysis and design are given to specify the proposed ETT approach with a real implementation.

1. 서론

21세기 정보기술의 초점은 시스템 통합이다. Data Warehouse(DW)는 과거 데이터의 분석에만 그치는 것이 아니라 데이터 마이닝, Enterprise Resource Planning(ERP), Supply-Chain Management(SCM), Customer Relation Management(CRM), Sales Force Automation(SFA), Computer Telephony Integration(CTI) 등 다양한 응용 정보기술과 접목되면서 21세기 정보기술의 중심으로 자리잡고 있다. 각 기업은 데이터 마이닝기법을 이용한 수요예측, 고객분석등 다양한 기법을 접목하고 있으며 DW는 기반기술로서 중요하게 대두되고 있다. Meta 그룹에 따르면 500명의 미국 기업의 각 정보기술전문가를 대상으로 한 설문조사에서 95%이상이 DW를 구축할 예정으로 있다고 한다. 각 벤더들도 이러한 흐름에 맞추어 각기 제품을 출시하고 있다. SAP Korea의 경우는 SAP 고객관리시스템에 DW 기반의 SAP 세일즈라는 솔류션을 가지고 공급하고 있으며 한국후지쯔도 세일즈포스비젼이라는 제품으로 DW와 그룹웨어와의 연동을 제공하면서 제품을 출시하고 있다. 유통업을 중심으로 DW 시스템은 국내에서도 개발사례가 증가하고 있다.

본 연구에서는 국내외의 많은 기업이 당면하고 있는 이질적 환경과 불규칙한 업무의 진행과 마감상황에서 DW 시스템의 근간이라고 할 수 있는 추출시스템을 효과적으로 설계하고 관련기술을 분석, 구체적으로 실제 구현함으로써 관련 DW의 요소기술과 연계 제시하고자 한다.

* 아주대학교 산업공학과

일반적으로 DW를 구축하는 기업은 기본적인 정보화가 진척된 기업이라 할 수 있다. 그러나 업무별로 개발된 시스템은 엑셀장표를 이용하는 수준에서부터 Frame기반의 기간계 시스템을 가지고 있는 기업까지 그 시스템이 다양하고 복잡하기 때문에 오히려 DW를 구축하는 걸림돌로 작용하고 있다. 기종간의 이질성과 일반 텍스트형식의 데이터처리와 DB와 DB간의 데이터 흐름은 그 내용 못지 않게 그들간의 통합과 성능(Performance)이 중요한 부분이다. 본 연구에서는 이기종간의 데이터를 처리하는 방법을 소개하고 데이터 추출시 발생하는 데이터의 정합성을 위한 스케쥴링과 Mapping 관계를 정의함으로 이기종의 환경에서 상호 이질적인 데이터를 어떻게 On-Line Analytical Processing (OLAP)이 요구하는 데이터마트(DataMart)까지 적재할 것인가를 논리적인 설계에서부터 실제 구축사례를 구체적으로 제시하고 분석 고찰하는 것이 그 목적이다.

2. 관련 용어 및 제안 ETT의 설계분석

2.1 데이터웨어하우스의 정의

DW란 수년간의 기업의 운영계 시스템에서 생긴 내부 데이터와 외부 데이터를 주제별로 통합하여 별도의 프로그래밍 없이 즉시 여러 각도에서 분석을 가능케 하는 통합시스템이다. 과거에도 정보창고에 대한 이론이 존재해 왔지만 오늘에 와서야 하드웨어의 발전과 비용이 저렴해 각 기업에서 ROI(Return on Investment)를 비교하여 얼마든지 DW 구축을 시도할 수 있는 상황이다.

2.1.1 데이터 웨어하우스의 기본 구성도

DW는 기간계 시스템과 별도로 구축하는 것이 일반적이며 데이터는 기간계 시스템에서 추출하여 DW 서버로 로드 된다. 로드 된 데이터는 다차원 모델링을 통하여 재 가공되며 OLAP 툴을 이용하여 분석 가공하여 사용자의 요구를 만족시킬 수 있다.

2.1.2 DW의 기술적 배경

CPU의 성능이 급격히 향상되고 메모리와 저장장치의 발전과 비용의 하락은 오픈시스템을 구축할 수 있게 하였다. 특히 오픈 환경을 지향하면서 메인프레임급 성능을 가지고 있는 컴퓨터가 등장하고 이는 대칭형프로세서(SMP), 클러스터(cluster), 초병렬 프로세싱(MPP)등 새로운 컴퓨터 구조의 등장에 힘입은 바 크다. 또한 새로운 운영체계도 이러한 기술적 배경이 되고 있다. 이에 따라 [그림1]에 나타난 바와 같이 Operating Data Store(ODS), ETT(Extraction Transformation Transportation)) 등은 DW를 구축하는 핵심적인 기술로 대두되고 있다.

2.2 다차원모델링

2.2.1 모델링 기법

본 연구에서는 다차원 모델링이 완성되어진 후 모델링이 원하는 데이터를 추출하여 최종 타겟 테이블까지 테이타를 로딩하는 ETT의 설계 및 구현을 제시한다. 다차원적 모델링 기법은 일반적으로 크게 두가지 방식으로 나눌 수 있는바 스타스키마 방식과 스노우플레이크 스키마 방식이 그것이다. 스타 스키마는 하나의 차원테이블에 여러 가지의 속성이 존재하는 방식이고 스노우플레이크 스키마는 각 속성마다 따로 차원 테이블을 만들어 주는 방식이다. 어떤 방식이 좋은지는 사용자의 요구에 따라 다르다. 간단한 다차원분석을 위해서는 스타스키마 방식이 좋고 복잡한 다차원 분석은 스노우플레이크 스키마 방식이 좋다. 이 외에도 다차원 모델링에서 생각해야 할 것은 퍼포먼스튜닝(Perfprmance Tuning)이나 사실 테이블의 파티션(partition), 미세한 사용자의 요구사항을 반영해야 하는 갖가지 기법들을 고려해야 한다.

2.3 ETT 시스템 설계

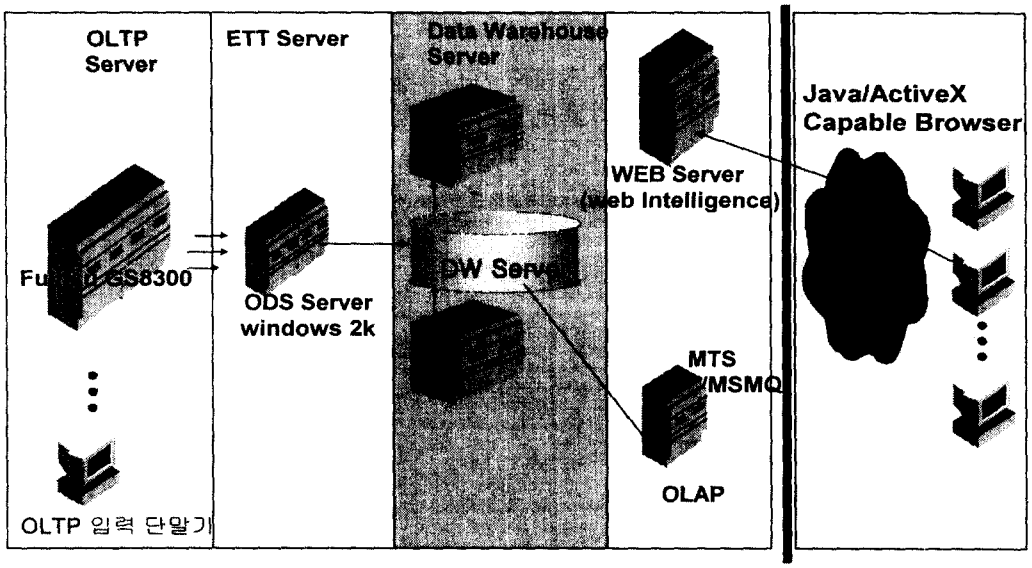

그림1 <데이터웨어하우스의 기술요소>

ETT(Extraction Transformation Transportation)는 데이터의 추출, 가공, 전송의 약자인데 데이터를 소스시스템에서 추출하여 DW에 로드시킨 상태에서의 정제작업까지 이른 전 과정을 말한다.

2.3.1 ETT의 형태

DW에서 필요한 최종 테이블은 실적데이터를 가지고 있는 사실테이블과 공통적인 데이터를 관리하는 차원 데이터들이 있다. ETT의 형태는 사실 테이블과 차원테이블을 어떠하게 설계하여 만들어 주느냐에 따라 시스템의 부하가 결정된다. 기간계 시스템에서 대부분의 차원테이블을 작성하여 DW쪽으로 추출한다면 DW의 역할을 그 만큼 간단해 지고 부하 또한 감소된다. 하지만 기간계 시스템의 부하를 무시할 수 없으므로 시스템 구현 시 기간계의 부하를 고려, trade-off관계를 잘 규명하여 결정하도록 하여야 한다. 일반적으로 기간계 시스템인 경우 실제 업무가 운영되는 곳이므로 시스템의 부하로 인한 업무의 지연을 잘 고려하여야 한다.

2.3.2 데이터의 변환

ETT시스템에서는 데이터의 변환과정이 필수적인바 이는 DW의 특정한 포맷, 사용자의 요구를 충분히 반영할 수 없는 데이터의 구조, 소스 데이터의 코드 불일치성이 존재하기 때문이다. 데이터정제는 크게 2가지 방법으로 이루어 진다. 첫째는 SAM 파일을 이용한 데이터의 정제작업이다. SAM 파일을 이용한 정제작업은 DB 의 특성에 따라서 그 성능이 좋아질 수 있으며 이기종의 데이터를 통합하기 위한 대안 일 수 있다. 하지만 일반적으로 Unix로 들어오는 SAM파일을 로드시키기 위해서는 여러 단계의 변환작업이 필요함으로 디스크의 낭비가 우려되고 프로그램의 개발시간이 길고 정확한 개발도 어려워 사용자의 요구조건이 바뀌면 유연하게 대응할 수 없다는 단점이 있다. 두 번째는 ODS를 이용한 데이터의 정제이다. ODS의 역할은 데이터의 가공, 변환, 정제작업을 용이하게 해주며 사용자의 요구조건에 신속하게 대응할 수 있게 해주며 사실 테이블에 문제가 생겼을 때 신속히 복구하게 해 준다. 하지만 이런 방법도 데이터가 급속히 증가할 경우 DB자체에서 처리하는 것은 많은 부하가 발생한다. 이런 경우에는 첫 번째 방법인 로드를 통하여 데이터를 정제하는 방법을 찾아야 한다.

2.3.3 ETT의 방식

ETT는 데이터를 전송하는 방법에 따라 다음과 같은 오프라인 방식과 온라인 방식으로 나눌

수 있다.

(1) 오프라인 방식

오프라인 방식은 기간계 시스템에서 파일 형태로 데이터가 내려오는 것을 뜻하며 이때 주로 SAM 파일을 이용하여 FTP를 사용하는 방법과 저장매체를 이용하여 데이터를 전송하는 경우로 볼 수 있다.

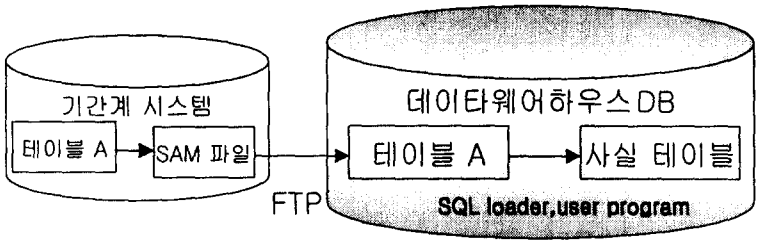

그림 2 <오프라인 추출방식>

[그림2]와 같은 방식은 데이터를 실시간으로 볼 수 없다는 단점과 FTP로 전송시에 전송 중 장애 발생시 장애유무를 찾아내기가 곤란한 단점이 있지만 시스템이 전체적으로 안정적이며 데이터의 유지보수가 간단해 진다는 장점이 있다.

(2) 온라인 방식

온라인 방식은 기간계 시스템에서 DW로 직접 데이터를 내리는 방식이다. 이럴 경우는 기간계 시스템과 동일한 테이블 구조를 DW쪽에서 만들어주고 변형이 필요할 경우는 SQL Script나 PL/SQL등을 사용하여 변형 및 정제한다.

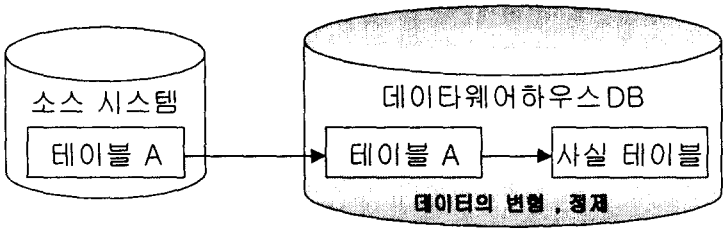

그림3 <온라인 추출방식>

(3) 디스크 공유방식

디스크 공유방식은 시스템에서 특정한 데이터가 바뀌면 로그 파일을 만들어 놓고 데이터 웨어하우스에서 주기적으로 체크해서 데이터를 가져오는 방식이다. 이러한 방식의 단점은 실시간으로 데이터의 추출, 변환, 정제, 요약작업까지 완전 자동화를 지향해야하고 에러가 발생할 시 복원기능에 특히 신경 써야 한다.

이와 같이 ETT 시스템은 DW의 가장 핵심적인 부분을 담당하고 있으며 OLAP 시스템이 구축되기 위한 가장 구체적이고 신속, 정확해야 한다. ETT시스템은 하나의 정형화된 방법을 제시하기가 힘들다. 구축을 하고자 하는 시스템의 성능과 기종에 따라 여러 가지 방식을 택할 수 있고 제공되는 유틸리티들도 있다. 하지만 데이터를 추출하고자 하는 방법은 위의 방법과 같이 3가지 정도로 분류할 수 있으며 이들간의 장점을 고려한 정형화된 추출 로직을 제시하는 것은 매우 유용하다고 할 수 있다.

3. 제안 시스템 설계 및 구현

3.1 제안 시스템 구조 평가

제안 시스템 구조는 FRAME 컴퓨터를 기반으로 하는 OLTP환경의 기간계 시스템과 업무영역별로 개발된 C/S시스템을 대상으로 한다. [그림4]와 같이 메인프레임에는 전국의 각지점과 영업소, 공장이 접속되어있어 경영실적과 근간이 되는 데이터들을 처리하고 있다. 또한 C/S환경으로 개발된 시스템들은 일일실적을 관리하기 위한 PC서버와 인사서버 회계서버 등으로 개발되어 있다. 전체적으로 시스템을 평가하여 보면 단위업무별로는 전산화가 상당히 잘 구축되어 있으며 관리 또한 잘 이루어지고 있다고 평가할 수 있다. 본 연구에서 구축하려고 하는 DW 시스템은 사례 연구지에 존재하는 모든 데이터들을 처리하여야 하기 때문에 DW를 구축하기 위해서는 시스템의 이질적 환경을 극복하고 단위업무별로 일정하지 않은 일일업무 마감시간을 고려하여 데이터의 추출을 결정하여야 하는 문제를 가지고 있다.

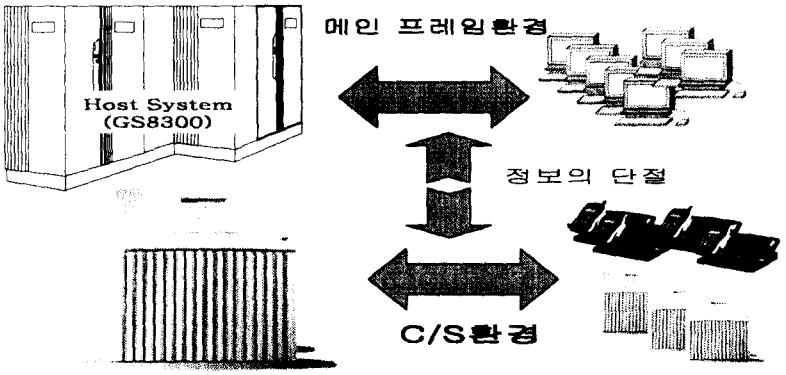

그림 4 AS-IS System Architecture

3.2 DW를 위한 시스템의 문제점

3.2.1 프레임기반의 OLTP 시스템

OLTP 시스템은 실시간 데이터들이 처리되는 시스템이다. 업무의 특성상 많은 데이터들이 처리되고 있으며 현재 사용중인 테이블에서 데이터를 추출하는 부분은 시스템의 에러와 실행시간(Performance)을 줄이기 위해 잘 설계되어져야 한다. 이와 같은 점을 고려하면서 프레임기반의 사례 연구지에서 기간계 시스템의 문제점은 다음과 같다.

(1) 일일업무처리량의 과다

일일업무처리의 과다는 그만큼 DW로 추출하기 위한 시스템의 유휴시간이 없음을 나타낸다. 이와 같은 문제점을 해결하기 위하여 본 연구에서는 CPU에 점유되어 지는 시간을 줄이기 위하여 데이테웨어하우스의 주제 영역별로 업무를 나누어서 병렬 처리하는 방법을 선택하였다.

(2) 운영체제의 이질성

사례 연구지의 시스템은 프레임 기반의 운영체제로 기동되고 있으며 데이터의 구성도 EBCDIC으로 구성되어 있다. 일반적으로 DW 시스템이 Unix 운영체제를 사용한다고 가정하면 유닉스 운영체제에의 데이터구성은 ASCII로 되어 있기 때문에 데이터의 추출을 위해서는 이와 같은 이질성을 해결해야 하는 문제점이 있다. 본 연구에서는 이러한 문제를 해결하기 위하여 코드 맵핑 테이블을 이용한 프로그램을 사용하여 이들간의 호환성 문제를 해결하도록 하였다.

3.2.2 업무별로 개발된 C/S기반의 시스템

사례 연구지의 업무단위별 시스템들은 시스템의 통합을 고려하지 않고 판매속보시스템, 인사시스템, 회계시스템, 장비관리 시스템등이 각기 개발되어 있으며 이들은 각기 MS_SQL, 오라클 , MS-Access등의 DBMS로 구성되어 있다. 이와 같은 경우 이기종 DB간의 데이터 이동이 문제가 된다. DBMS의 데이터 저장 방식이 서로 상이하고 또한 데이터의 추출이 가능하다고 할지라도 대량의 데이터가 내부 네트워크를 통해 서로 데이터를 주고받는 다는 것은

Performance를 떨어지게 하는 한 요인으로 작용한다. 본 연구에서는 Performance의 향상을 위하여 각 경우에 맞는 추출 알고리즘을 설계하였으며 상세한 내용은 ETT시스템 구현에 제시되어 있다.

3.2.3 공통코드의 불일치로 인한 데이터의 비정합성

업무별로 개발된 C/S 기반의 시스템과 기간계 시스템사이에서는 하나의 제품을 두고 서로 명명하는 상품코드의 체계가 동일하지 않다. 이러한 문제는 데이터의 정합 성을 떨어뜨리고 분석의 효율성을 떨어뜨리는 요인이 된다. 이를 해결하기 위하여 본 연구에서는 매핑 테이블과 PL/SQL을 이용하여 이들간의 상이한 코드체계를 정리하며 표준화하였다.

3.3 추출로직 설계

3.3.1 추출 대상선정

본 연구를 위한 업무 대상은 영업, 인사, 회계영역을 대상으로 하였으며 물류의 일부 데이터도 포함하여 진행하였다. 또한 시스템화 되어 있지 않고 비정규화된 내외부데이터도 추출의 대상에 포함한다.

3.3.2 변환설계서 및 ERD설계

추출대상으로 선정된 데이터들은 서로 각기 다른 DB에 존재한다. 영업데이터의 경우는 프레임 컴퓨터에 존재하고 회계 인사의 경우는 SQL Server에 존재한다. 이렇게 각기 다른 시스템에서 데이터를 추출 할 때는 DW 모델링을 통하여 추출 데이터를 결정하는 것이 좋다. DW모델링을 하기 위한 간단한 단계는 각각의 사용자가 보기를 원하는 데이터를 조사해야 하며 업무에 사용하는 각종 장표들이 어떤 데이터를 이용해서 보고되며 업무에 사용되는지를 조사해야 한다. 이러한 분석이 끝나면 [그림5]와 같은 DW 모델링을 해야하며 여기에서 문제점이 실제 사용자가 필요로 하는 데이터들은 실제 원천 데이터 즉 하나의 테이블이나 하나의 데이터

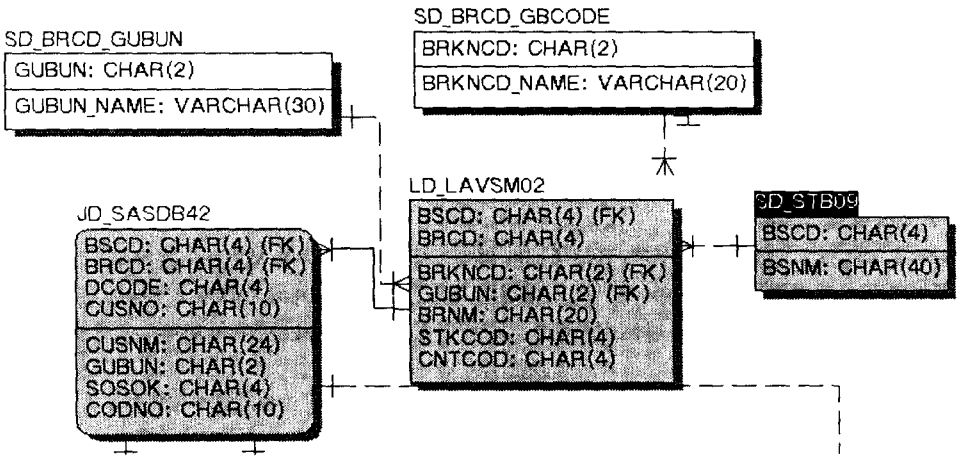

그림 5 ERD

베이스에 존재하지 않는다. ETT 시스템에서는 이러한 점을 해결해야 한다. 본 사례연구에서는 [그림5]의 DW모델링을 근간으로 하여 [그림6]변환설계서를 작성하였다. 변환설계서는 DW모델링에서 필요로 하는 데이터를 만들기 위해 하나 또는 그 이상의 칼럼들을 추출하여 컬럼간의 조합으로 DW에서 원하는 데이터를 제공해 줄 수 있다.

[그림6]은 영업의 일부인 방문판매 부분의 변환 설계서를 작성한 것이다. 추출의 최종 테이블과와 추출에 사용되는 원천데이터는 일반적으로 정의된 규칙은 없으나 추후 데이터의 검증과 기간계 시스템의 부하를 방지하기 위하여 ODS로 데이터를 기간계 시스템과 동일하게 추출

하여 이 추출된 데이터를 원천데이터로 규정하는 것이 일반적이다. 그 이유는 기간계 시스템에

TARGET 테이블 한글명	주문_공통	TARGET 테이블영문명	JF_SASDB43	예상건수(크기

SEQ	TARGET			SOURCE				주기
	컬럼한글명	컬럼영문명	데이터타입	테이블영문명	컬럼한글명	컬럼영문명	데이터타입	
1	대리점공급요청일자	ASKYMD	CHAR(8)	SASDB43	대리점공급요청일자	ASKYMD	NUMERIC(8)	일
2	대리점주문일시	JUILSI	CHAR(12)	SASDB43	대리점주문일시	JUILSI	CHAR(12)	주
3	대리점코드	DCODE	CHAR(4)	SASDB43	대리점코드	DCODE	NUMERIC(4)	
4	본사전산처리일시	COMILSI	CHAR(12)	SASDB43	본사전산처리일시	COMILSI	CHAR(12)	
5	일련번호	SEQ	CHAR(3)	SASDB43	일련번호	SEQ	NUMERIC(2)	
6	주문구분	JUGU	CHAR(2)	SASDB43	주문구분	JUGU	VARCHAR(2)	
7	주문일자	YMD	CHAR(6)	SASDB43	주문일자	YMD	NUMERIC(8)	
8	지점확인일시	CONILSI	CHAR(12)	SASDB43	지점확인일시	CONILSI	CHAR(12)	
9	진행상태	STATUS	CHAR(1)	SASDB43	진행상태	STATUS	VARCHAR(1)	
10	부서코드	BSCD	CHAR(4)	JD_SASDB42	부서코드	BSCD	CHAR(4)	
11	지점코드	BRCD	CHAR(4)	JD_SASDB42	지점코드	BRCD	CHAR(4)	
12	사업자번호	CUSNO	CHAR(13)	JD_SASDB42	사업자번호	CUSNO	CHAR(13)	

그림 6 변환설계서

서 DW가 원하는 데이터를 만들어 추출하기 위해서는 수많은 Join 관계가 형성되고 데이터베이스상의 Temporary Size를 필요로 하기 때문에 오히려 DW의 성능향상만을 고려하다 보면 실제 실시간으로 운영되고 있는 기간계 시스템의 성능저하를 초래할 수 있기 때문이다. 또한 [그림6]과 같이 부서코드, 지점코드, 대리점코드부분은 실제 소스테이블에 존재하지 않는 것으로 [그림6]의 변환이 발생하기 전에 이미 JD_SASDB42라는 테이블이 존재하여야 함을 알 수 있다. 이와 같은 이유는 기존의 시스템을 독립적으로 개발한 이유와 OLTP상의 데이테베이스 모델링과 DW 상에서의 데이터베이스 모델링의 방법적인 차이에서 발생할 수 있다. 일반적으로 OLTP 시스템에서는 데이터베이스 모델링을 정규화 시키는 방향으로 설계하지만 데이터 웨어하우스에서는 비정규화된 모델링을 요구하기 때문에 이러한 문제가 발생한다. ETT시스템 개발에서 제시하고자 하는 핵심적인 부분이 이와 같이 이기종 환경에서의 추출흐름을 결정하는 것과 모델링의 차이에서 발생하는 추출시의 스케쥴링을 결정하는 것이다.

3.3.3 추출흐름 순서결정을 위한 로직 설계

추출 흐름의 로직이 필요한 이유는 기간계 시스템의 업무별 마감시간과 DW모델링에서 처럼 먼저 추출되어져야 할 것과 나중에 추출되어야 할 테이블들이 존재하기 때문이다. 그 이유는 정규화된 테이블들에서 비정규화된 DW 테이블들을 만들기 위해서는 기준테이블과의 조인이 필요하기 때문이다. ETT시스템에서의 추출방법은 오프라인 방식과 온라인방식으로 크게 나눌 수 있음을 밝혔다. 그러나 이러한 방법은 양단간 장단점을 가지고 있기 때문에 두 가지의

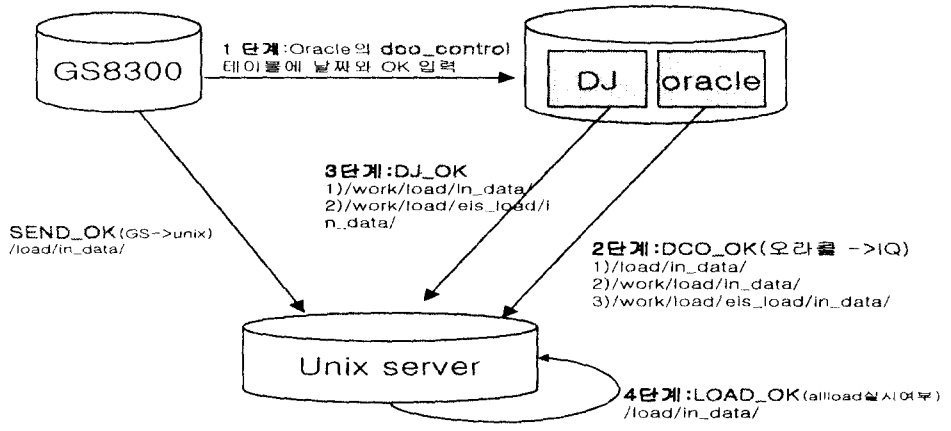

그림 7 flag를 이용한 추출흐름제어

방법을 혼합한 방법을 제시하고자 한다. 기간계 시스템에서 FTP로 SAM 파일을 해당 시스템에 전송하는 부분은 오프라인 방식이지만 SAM 파일이 전송된 이후에는 모든 추출 알고리즘이 자동화 되도록 설계하였다. 구현 된 DBMS는 SybaseIQ를 사용하기 때문에 최종적으로 데이터가 적재되어야 하는 곳은 SybaseIQ 이다. 시스템의 문제점에서 제시되었던 것처럼 각기 업무별로 개발된 시스템으로 인해 데이터는 코드의 불일치성과 물리적으로 서로 다른 곳에 저장되어 있어 추출순서의 문제가 중요한 이슈이다. 적용사례 연구지에서 설계한 방법은 [그림7]과 같이 flag를 이용한 자동 추출방법이다. 서로의 운영체제가 틀리고 데이터의 저장 타입이 틀리기 때문에 호완성이 없어서 추출완료 여부를 소스시스템의 로그를 이용해 확인하기가 곤란하다. 이러한 점을 해결하기 위해서는 추출되는 시스템에서 데이터의 송신이 종료된 후 종료 확인을 대상 시스템에 로그파일로 남겨주는 방식이다. 이 방식은 통신수단으로 FTP를 이용하며 데이터베이스의 테이블을 이용하기 때문에 이질적인 환경과 데이터 타입과 상관없이 작업 순서를 제어 할 수 있는 효과가 있다. 프레임컴퓨터가 ODS서버에 전송하는 데이터들은 제품 코드 체계의 클린징 작업을 위한 코드성 테이블들이다.

[그림7]에서와 같이 프레임컴퓨터는 자신의 전송이 끝나면 ODS서버에는 당일날짜와 전송이 완료 되었다는 의미로 DCO_CONTROL이라는 오라클 테이블에 로그를 남기게 된다. ODS서버에서는 오라클의 Job-QUEUE를 이용하여 작업이 항상 수행되고 있으나 프레임컴퓨터에서 로그를 확인하여 로그가 있을 경우만 해당업무를 수행하게 되어 있다. 오라클의 Job-Queue처럼 운영체제에도 작업을 시간을 중심으로 제어할 수 있는 기능들이 있다 Unix에는 Crontab, AT이라는 기능이 있으며 NT에는 AT이라는 기능이 있다. 본 사례연구에서는 이러한 OS의 기능과 DBMS에서 제공하는 Job-control을 이용하여 해당 작업을 계속 수행하되 제어 Flag가 존재할 경우만 실제 업무를 수행하도록 설계하였다. 이러한 플래그를 이용한 자동추출 로직을 사용하면 업무의 제어를 통하여 추출 시 발생하는 순서제어의 문제를 자동으로 해결 할 수 있다.

3.4 ETT 시스템 구현

[그림8]은 종합적인 ETT시스템의 개략도이다. [그림8]의 ETT시스템 구현은 DW 전용 DBMS인 SybaseIQ와 ODS에서 데이터의 정제용으로 Oracle을 사용하였으며 추출도구로 Linkexpress, Data Junction, 사례연구지에서 개발한 프로그램을 이용하였다. 운영체제는 AIX를 사용하는 프레임컴퓨터와 Windows2000을 사용하는 ODS, Solaris5.7을 사용하는 DW서버가 있다. 사례연구지에서 개발한 언어는 쉘프로그램과 C++5.0을 이용하였다.

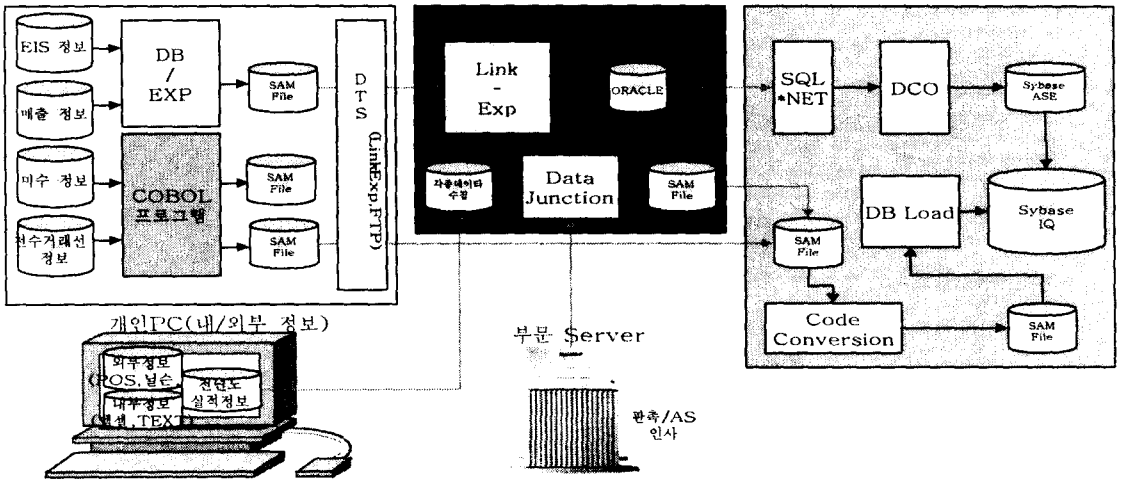

그림 8 ETT 시스템 Architecture

3.4.1 이기종데이터의 극복방안

(1) ODS의 해결방법

ODS에서는 추출시 링크익스프레스라는 툴을 사용하였다. 링크익스프레스는 NT기반에서 EBCDIC코드를 ASCII로 변환하여 오라클에 자동으로 적재하여 주는 역할을 하는 툴이다.

또한 DataJunction이라는 툴을 이용하여 SQL server와 엑셀데이터를 추출하여 DW서버로 적재하도록 하였다. 일반적으로 추출도구를 사용할 경우는 작업이 편리하다. 그러나 이러한 추출도구의 단점은 각 기업이 처해있는 다양한 환경에서 표준화시켜 사용할 수 없다는 것이다. 각각의 추출도구들은 DBMS에 종속되어져 있으며 특히 sybaseIQ기반의 DW시스템에서는 데이터베이스의 특성이 레코드단위의 저장과 별도의 인덱싱테이블을 가지고있는 것이 아니라 컬럼단위의 저장구조를 가지고 있기 때문에 추출도구를 이용할 경우 속도의 저하를 가져올 수 있다.

(2) DW서버

코드의 변환을 위하여 프로그램을 개발하여 데이터를 변환하였다. C환경으로 개발하였으며 변환에 관계되는 환경파일의 제작을 위해 shell 프로그램을 이용하였다.

위의 [그림8]에서와 같이 기간계 시스템인 프레임 컴퓨터에 저장되어 있는 데이터를 SAM 파일의 형태로 유닉스 서버에 FTP를 이용하여 전송한다. [그림9]와 같이 전송된 파일은 EBCDIC 코드의 체계로 되어있으므로 이 파일을 ASCII 파일로 전환할 필요성이 있다. 마지막 단계로 변환된 파일을 SybaseIQ 에 저장하기 위하여 컬럼과 컬럼 사이에 구분자를 삽입하여 최종파일을 만든다.

3.4.2 데이터 추출 스케쥴링 시스템의 구현

추출시스템이 해결해야 할 문제는 기간계 시스템에서 공통코드부분을 오라클에 저장하여 클린징한 후 DW 서버에 저장시키는 것과 독립적으로 개발된 각각의 시스템에서 정제과정을 거쳐 DW 시스템에 저장해야 하는 것이다. 각각의 독립시스템에서 데이터를 추출할때는 일정한 순서를 가지고 추출되어져야 한다. 이를 해결하기 위한 추출로직은 이전에 간략히 제시한바와

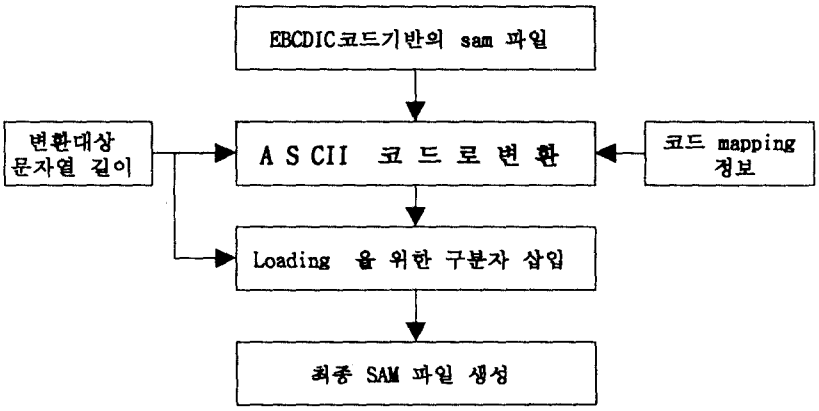

그림 9 변환 흐름도

같이 크게 오라클에서의 스케쥴링, DW 서버에서의 스케쥴링, 그리고 NT 기반의 독립시스템의 데이터 추출 스케쥴링으로 나눌수 있다.

첫째로 ODS인 오라클은 프레임 컴퓨터에서 데이터를 받아서 클린징을 하는 작업이 주 업무이다. 이때에 전송이 완료되지 않은 상태에서 작업이 수행된다면 조인관계가 불투명하게 처리되고 데이터의 정제가 이루어 질 수 없다. 이와 같은 문제를 해결하기 위하여 [그림10]과 같은

추출로직이 필요하다. [그림7]과 [그림10]은 오라클의 JOB QUEUE를 이용하는 스케쥴링 방법을 보여준다. 특히 [그림10]에서와 같이 일정한 시간간격을 두고 수행되는 작업은 [그림11]과 같은 작업완료 로그를 확인한후 수행될 수 있다. 둘째로 DW 서버에서의 스케쥴링이다. 오라클의 작업 완료유무를 확인하기 위하여 SyabseIQ의 DCO 기능을 이용하여 [그림10]의 작업완료 로그를 확인하고 자동작업을 위하여 유닉스의 Crontab기능을 이용한다. crontab은 OS에서 제공해주는 스케쥴링 기능으로 일정 시간 간격으로 프로세스를 작동시켜주며 작동시 작업완료 유무를 먼저 확인하고 확인되었을 경우에만 실제 작업을 수행하도록 한다. 이 외에도 윈도우 NT기반의 시스템에서는 OS에서 제공해 주는 AT 기능을 이용하여 이전작업의 완료 flag를 확인하여 작업을 수행한다. 본 연구에서 제시하고자 하는 추출시스템의 자동화 스케쥴링이란 운영체제에서 제공해 주는 스케쥴링 기능과 flag를 이용하여 조합한 시스템이다. 이러한 기능을 이용하여 이전작업과 이후작업을 통제할 수 있으며 기간계 시스템의 추출에서 데이터마트까지 데이터를 원할하게 적재할 수 있다.

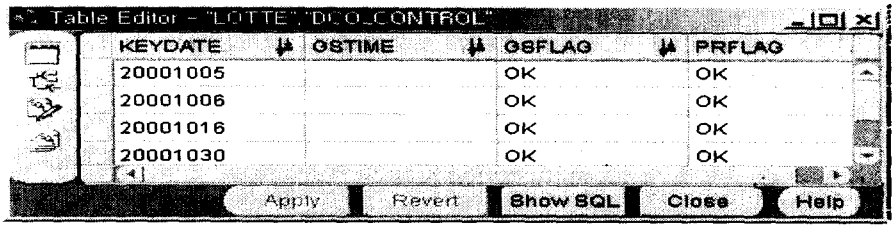

그림 10 프레임컴퓨터의 전송확인 창

4. 기대효과

국내의 많은 기업이 고비용 저효율의 문제에 직면해 이 있다. 실제 많은 업무에서 전산화가 이루어져 있음에도 불구하고 고비용 저효율의 문제는 상존하고 있다. 이러한 문제를 모니터링 할 수 있는 방법이 DW를 구축하는 것이라 생각된다. 앞으로도 많은 기업들이 DW를 구축할 것으로 판단되어 진다. 사례연구에서 제시된 것처럼 데이터의 추출은 DW 시스템에서 매우 중요한 역할을 한다. 이러한 맥락에서 본 연구의 기대효과를 정리해 보면 다음과 같다.

본 사례연구를 통한 기대효과는 DW 시스템에서 매우 중요한 부분인 ETT시스템을 실제적인 기업환경에서 구현한 것으로 볼 수 있다. 대부분의 기업이 직면하고 있는 기종간의 이질성으로부터 새롭게 모든 업무 시스템에 근간으로 자리매김하고 있는 DW의 구축은 이러한 기종간의 문제를 해결하고 추출의 자동화를 요구하고 있다. 본 연구의 구현사례를 통하여 이질적인 환경에서의 자동화된 데이터 추출이 가능할 것으로 판단되어 지며 다양한 추출경로를 통하여 시스템을 구축함으로써 최적의 시스템 성능을 유도할 수 있을 것으로 기대된다.

5. 결론

5.1 연구 요약

국내의 대부분의 기업들은 상당한 부분에서 각기 다른 운영체제와 데이터 타입을 가진 시스템을 운영하고 있다. DW 시스템은 전사적인 데이터를 이용하여야 하고 그러기 위해서 당면하는 문제가 이기종의 데이터 처리를 위한 로직설계이다. 본 연구에서는 이러한 문제를 해결하기 위하여 코드변환을 제공해 주는 툴을 사용하여 일부 해결하였으며 시스템의 최적의 퍼포먼스(Performance)를 위하여 코드 맵핑을 이용한 프로그램을 작성하여 해결하였다. 또한 업무단위별로 개발된 시스템에서 데이터의 추출을 시도하면 제품코드의 불일치성으로 인한 변환과 추출순서가 이슈가 되었다. 이와 같은 문제는 본 연구에서 제시한 Flag를 이용하여 순서제어를

할 수 있으며 추출시스템의 일반적인 추출방법에서 장점을 취합하여 운영체제와 DBMS에서 제공해주는 스케쥴링 기능으로 추출의 자동화 로직을 제시하였다. 이렇게 자동화된 로직은 기간계 시스템 및 독립적으로 개발된 업무별 시스템에서 최종 데이터 마트에 적재되기까지 일괄적으로 처리될 수 있다.

5.2 향후연구과제

데이터 웨어하우스는 이제 기존의 시스템인 ERP를 비롯하여 많은 시스템의 근간이 되는 시스템으로 발전하고 있다. 시스템의 발달과 업무의 전산화는 DW가 처리해야할 데이터의 량도 기하급수적으로 증가시키고 있으며 또한 기업의 입장에서는 이러한 수많은 데이터를 어떻게 잘 보관하며 이용할 수 있느냐가 중요한 이슈로 대두되고 있다. 이러한 시점에서 본 연구 주제에 대한 향후 방향은 DW는 시스템의 성능과 매우 밀접한 관계가 있다. 일반적으로 데이터를 처리하는 방법도 기존의 RDBMS에서는 row 단위로 처리를 하지만 데이터 웨어하우스에서는 컬럼 단위로 데이터를 처리해야 성능이 향상되는 경우가 많이 있다. 데이터 웨어하우스 시스템이 정상적으로 안정되게 작동하기 위하여서는 시스템 퍼포먼스를 어떻게 측정하여 최적의 설계를 하느냐가 중요한 관건으로 판단되어지고 앞으로 시스템 튜닝과 ETT 로직에 대한 튜닝 방법에 대한 연구가 필요시 된다.

6.. 참고문헌

[1] 문송천 "Dataware 설계론",아이포스트 ,1999

[2] 장동인, "실무자를 위한 데이터웨어하우스",대청출판사,1999

[3] 조재희,박성진," 데이터웨어하우스의 효과적 활용기 법 OLAP Technology",시스마지식경영연구소,2000

[4] W.H.Inmon, "Building the Datawarehouse" ,홍릉과학출판사, 1997

[5] Larry P.English,"Data Warehouse and Business Information Quality",wiley,2000

[6] Neilson Thomas Debevious,"The Data Warehouse Method",prentice hall PTR, 1999

[7] W.H.Inmon, "Data Warehouse Performance", wiley,1999

[8] Mike Loukids, "System Performance Tuning", O'Reilly,1996

[9] Alex Berson,Stephen Smith, Kurt Thearling , "Building Data Mining Allication for CRM". Mc Graw Hill. 2000