

데이터 웨어하우스 ETT 도구들의 평가 및 검증

김기운*, 서용무*

A Comparative Test of ETT Tools for Data Warehousing

Kim, Giun, Suh, Yongmoo

Many enterprises continue to have an interest in the usage of new information technologies to gain a competitive advantage. In particular, their interest in the data warehouse and the data mining reveals the aspect of such a trend. Although lots of vendors announce a variety of tools for data warehousing, many a enterprise have a difficulty in building a robust data warehouse due to the lack of the ability of selecting an appropriate data warehouse technology options. Therefore, this study presents some evaluation factors, evaluation methods, and evaluation results about ETT tools, mainly in terms of a comparative test for the current available data warehousing ETT tools. Also, this paper suggests some guides about choosing the right ETT tools.

* 고려대학교 경영학과

I. 서론

효과적인 의사결정 지원 시스템을 구축하기 위해서는 전사적인 데이터는 3가지의 필요한 요건에 부합할 수 있어야 한다[Kelly, 1994]. 첫째로 데이터는 조직 내부의 다른 전사적인 데이터와 통합되어야만 한다. 둘째는 외부 데이터와의 통합 등을 통하여 데이터의 내용이 풍부해져야만 한다. 셋째는 데이터의 가용성(availability)이 높아야 한다. 기업들은 이러한 요건들을 충족할 수 있는 적절한 하드웨어와 소프트웨어의 플랫폼을 고려할 필요가 있다. 이러한 노력의 일환으로 기업들은 의사결정 지원을 위한 필수 요소로 데이터 웨어하우스와 OLAP(On-Line Analytic Processing)의 플랫폼을 고려하고 있다. 그러나, 많은 기업들은 자사의 데이터 웨어하우스 구축 목적에 부합하는 적절한 데이터 웨어하우스 도구를 선정할 수 있는 능력이 부족하기 때문에 데이터 웨어하우스를 구현하기 위하여 이용될 수 있는 여러 기술 대안들을 평가하고 선택하는 문제에 있어 기업들은 상당한 어려움을 겪고 있다. 더욱이 Inmon[1992]는 데이터 웨어하우스에 관한 그의 책에서 정보를 데이터 웨어하우스로 추출(extraction), 정제(cleansing), 적재(loading)하는 일은 상당한 시간을 소요한다고 지적하였다. 평균적으로 데이터 웨어하우스를 구축하는 노력의 80%가 이들 업무에 투입된다[Barquin & Edelstein, 1996]. 따라서, 성공적인 데이터 웨어하우스를 위한 올바른 플랫폼과 기술 대안을 선택하는 것은 매우 중요하다고 할 수 있다. 본 연구에서는 데이터 웨어하우스를 구축하기 위하여 요구되는 정보기술에 대한 평가항목을 제시하고, 그 평가항목에 대한 평가방법, 평가과정 및 그 결과에 대한 실제 검증 내용을 소개한다.

II. 데이터 웨어하우스의 특징

2.1 데이터 웨어하우스의 정의

“데이터 웨어하우스”라는 용어는 수년간 정보 기술 주변에 있어 왔던 개념의 또 다른 표현이라 할 수 있다. 일 예로, IBM은 정보 웨어하우스(Information Warehouse)라는 용어를 오래 전부터 사용하였다. 1990년 데이터 웨어하우스라는 용어를 처음 사용한 William Inmon이 그 개념을 정립한 이래로 그의 견해와 크게 다르지 않게 정의되어 왔다. 데이터 웨어하우스에 대한 기존의 정의를 정리해보면 <표 1>과 같다.

기존 정의들의 핵심은 바로 “의사결정을 지원하기 위한 데이터베이스”라는 것이다. Kelly[1994]는 데이터 웨어하우스의 주요 특징을 다음과 같이 분류하였다: 1) 전형적으로 데이터 웨어하우스의 원천이 되는 기존의 운영시스템과 분리되어 구축되며 운영 데이터베이스에 의해 지원되는 OLTP(On-Line Transaction Processing) 응용시스템의 기능과 성능 요건과는 전혀 다른 OLAP 응용시스템을 지원한다. 2) 업무 사용자들이 알고 싶어하는 것을 질의에 의해 이용 가능하게 해준다. 3) 주제 지향적으로(subject-oriented) 저장되고 기업의 표준 모델을 기반으로 통합되어 있다. 4) 데이터에 시간을 부여하여(timestamped) 추세 분석과 같은 다양한 업무 목적에 이용할 수 있도록 데이터를 저장한다.

데이터 웨어하우스가 주제 지향적이라는 것은 과거에 트랜잭션 중심이었던 데이터가 주제 중심으로 재구성되어 의사결정을 지원한다는 의미이며, 여러 데이터가 하나로 통합된다는 것은 다양한 원천(source)에 걸쳐 흩어져 있는 데이터들이 단일 원천으로 통합되며 이 과정에서 데이터의 형식(예를 들어, 속성의 이름, 자료의 표현, 도량형의 단위)들을 일관성이 있도록 통

<표 1> 데이터 웨어하우스의 기존 정의

출 처	정 의
Inmon & Hackathorn [1994]	의사결정을 지원하기 위한 주제 지향적이고 통합되었으며 시계열적인 비휘발성(갱신되거나 삭제되지 않는) 데이터의 집합
Whitten et al. [1994]	여러 데이터베이스 및 전통적 파일로부터 추출된 데이터의 저장소
Poe [1996]	의사결정지원시스템의 기초로 사용되는 읽기 전용의 분석적 데이터베이스
Saylor & Bansal [1995]	의사결정을 지원하기 위해 최적화된 데이터베이스 또는 데이터베이스들의 집합
Widom et al. [1995]	질의 및 분석을 위해 활용할 수 있는 통합된 정보의 저장소
Strange, Burton, Gartner Group[1994, 1995]	하나의 아키텍처이며, 아주 중요한 비즈니스 의사결정을 내리기 위해 정보를 필요로 하는 사용자에게 데이터를 전해주기 위한 입체적인 전략
EDS [1995]	특별히 의사결정 지원을 위하여 설계된 주요 비즈니스 정보 데이터의 모음
홍의기[1996, 1997]	기업의 내·외부 자료 원천으로부터 분석 및 의사결정에 필요한 현재 및 과거의 데이터를 자동적으로 추출하여 의사결정 환경에 맞게 최적화하여 통합되어 변환된 주제 지향적인 읽기 전용 데이터베이스
Labio et al[1997]	분산되어 있고 자율적(autonomous)이고 이질적인 정보 원천으로부터 통합된 정보의 저장소
박천승[1997]	기업의 운영계 시스템에 누적된 내부의 데이터와 외부 데이터를 따로 의사결정을 지원하도록 가공하여 현재 및 과거 이력, 요약 데이터의 형태로 주제별로 통합하여 저장하는 분석용 데이터베이스

<표 2> 전통적 데이터베이스와 데이터 웨어하우스 비교

	전통적 데이터베이스	데이터 웨어하우스
주요 목적	일상 운영 업무의 자동화	장기적 의사결정지원 (전략적)
시간 범위	현재	과거, 근래
재충전(Refresh)	즉시	주기적으로
보유 데이터	상세한 데이터만	상세 데이터, 요약 데이터
데이터 유형	동질적인 현 시점의 데이터	방대한 양의 이질적인 근래 및 역사적 데이터
데이터 특성	갱신 가능, 기능 지향적	사용자 갱신 불가능, 주제 지향적, 통합적
주요 사용자	운영 업무 담당자	관리자 및 임원

일한다는 의미이다. 또한 데이터 웨어하우스로 들어오는 데이터는 다단계 요약정보로 변환되며 5년 내지 10년 간에 걸쳐 누적되어 목시적으로나 명시적으로 시간 항목을 가진 스냅샷(snapshot)¹⁾ 데이터의 성격을 띠게 되며, 갱신없이 읽기 전용으로 계속 유지된다. 이러한 데이터는 기존의 운영시스템의 성능에 영향을 미치지 않게 하기 위하여 별도의 데이터베이스에 저장한다.

본 연구에서는 Inmon[1992]의 정의를 기초로 기존의 정의들을 종합하여 “데이터 웨어하우스는 운영계 시스템에서 발생한 데이터(operational data), 외부 데이터(external data), 또는 별도 입력 데이터를 일정한 주기별로 추출 및 변환을 하여(ETT), 사용자 요구사항을 주제별로 통합하고, 일정 시점(일, 월, 분기, 년 등)별로 수년간 보관한 데이터(historical data)를 최종 사용자가 여러 각도에서 분석(multi-dimensional analysis), 조회(query) 및 출력(report)을 할 수 있도록 하

1) 어떤 테이블로부터 임의의 순간에 조건에 해당하는 값을 갖는 새로운 테이블을 추출하여 저장한 테이블

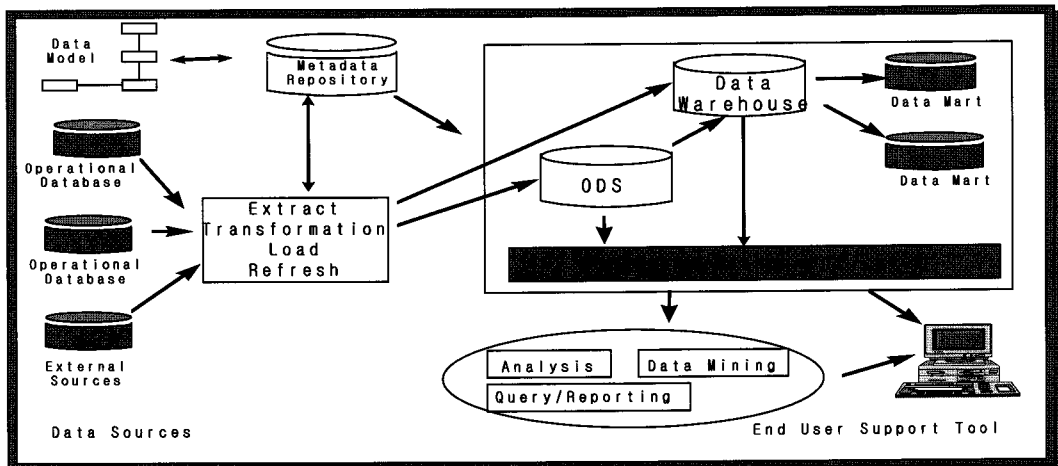
여 기업의 주요 의사결정을 지원하는 통합시스템"이라고 정의한다.

위와 같은 여러 정의를 바탕으로 데이터 웨어하우스의 개념을 명료화하기 위하여 전통적 데이터베이스와의 상호간 특징을 비교하면 <표 2>와 같다[김기운외, 1999]. 전통적 데이터베이스는 일상적인 데이터의 거래 즉 OLTP를 지원하도록 구축되어, 갱신 거래가 발생하면 그에 해당하는 데이터 항목의 과거 값들을 잃어버리게 된다. 반면, 데이터 웨어하우스는 여러 가지 의사결정을 지원할 수 있도록 데이터 분석 등에 초점을 둔 OLAP를 지원하도록 구축된다. 즉 전통적 데이터베이스의 데이터 생명 주기는 일시적이라면, 데이터 웨어하우스는 비교적 방대한 양의 역사적 데이터를 토대로 장기적인 의사결정 및 분석을 적시에 지원하는 것이 목적이므로 복잡한 질의 요청에 대한 신속한 응답을 위하여 최근의 데이터 뿐 아니라 이력 정보에 대한 요약된 데이터가 물리적으로 저장된다는 차이점을 가진다. 따라서 데이터 웨어하우스의 크기는 전통적 데이터베이스에 비해 상당히 크다. 또한 전통적 데이터베이스의 사용자들은 데이터 항목을 읽거나 갱신하기 위하여 응용시스템을 통하여 데이터베이스에 접근을 하는 반면에, 데이터 웨어하

우스의 사용자들은 그들이 원하는 정보를 검색하기 위하여 OLAP 도구를 통하여 데이터에 접근을 한다. 전통적 데이터베이스는 상세한 데이터를 데이터 웨어하우스에 제공하며 데이터 웨어하우스는 여러 정보 원천으로부터 제공된 정보를 통합한다.

2.2 데이터 웨어하우스 아키텍처

아키텍처라함은 어떤 시스템 혹은 제품의 전반적인 설계를 위한 프레임워크를 제공하는 규칙 또는 구조의 집합이다[Poe, 1996]. 데이터 아키텍처는 데이터가 시스템 전반에 걸쳐 어떻게 이동하고 기업 안에서 어떻게 활용되는 가를 식별하고 이해할 수 있도록 돕는 프레임워크이다. 데이터 웨어하우스 아키텍처는 그 주요 구성요소로 의사결정 지원을 위해 사용되는 읽기 전용 데이터베이스와 이들을 통합하여 형성되는 하나의 대용량 데이터베이스가 있는데, 다른 시스템과 차별화 할 수 있는 자신만의 독특한 특징을 갖고 있다. 그 동안 학계 및 업계에서의 쟁점 중의 하나는 데이터 추출과 통합의 자동화라 할 수 있다. 본 절에서는 <그림 1>과 같이 원천 데이터베이스 시스템, 데이터 모델, 자료의 추출·변형·전



<그림 1> 데이터 웨어하우스 아키텍처

송(ETT; Extract · Transform · Transport), 메타 데이터 저장소, ODS, 목표(target) 데이터베이스, 데이터 마트, 데이터 웨어하우스 관리층, 최종 사용자를 위한 접근 도구 등을 포함하는 아키텍처를 설명하고자 한다.

(1) 원천시스템

데이터 웨어하우스는 내부 운영 시스템과 외부 데이터베이스로부터 가져온 데이터를 사용하여 구축된다. 데이터 웨어하우스를 위한 원천 데이터는 여러 개의 컴퓨터 상에 존재할 수 있는 IMS, DB2, VSAM 등과 같은 운영 데이터베이스와 파일 시스템으로부터 추출된다. 데이터 웨어하우스는 기존의 여러 원천으로부터 데이터를 추출, 정제하고 데이터 웨어하우스로 그 데이터를 적재하는 기능을 요구한다.

(2) 데이터 모델

데이터 웨어하우스는 데이터 모델, 데이터 정의, 물리적인 데이터베이스 구조에 의해 예시될 수 있는 포괄적이고 일관성 있는 전사적 데이터 아키텍처를 기반으로 한다. 전사적인 데이터 모델은 응용 시스템 전반을 비교할 수 있도록 해준다. 공통의 데이터 정의는 운영 시스템, 데이터 웨어하우스, 사용자 디렉토리에 있는 데이터 요소들을 설명(description)해 준다. 데이터 웨어하우스를 위한 데이터베이스 설계는 시간 요소, 요약, 유도(derivations) 요소를 포함할 수 있도록 비정규화된 설계가 허용된다. 즉 사용자의 이해를 용이하게 하고 데이터 웨어하우스의 성능을 위하여 어느 정도의 중복 구조가 추가되고 외부 데이터도 포함된다. 데이터 웨어하우스는 읽기 전용이며 질의, 보고를 위해 최적화되고 데이터의 이해를 돕기 위하여 단순화된다. 따라서 데이터 웨어하우스를 위한 설계는 거래 처리를 위한 설계와는 다르다. 거래 처리 데이터베이스는 정규화된 구조를 사용하는 반면에 데이터 웨어하우스는 업무 사용자의 관점을 반영하는 방

식으로 데이터를 조직하는 비정규화된 데이터 구조 및 미리 조인된 데이터 구조를 포함한다. 대표적인 기법에는 "star 스키마²⁾" 혹은 "snowflake 스키마³⁾"가 있다.

(3) ETT(Extract, Transform & Transport)

ETT는 원천 시스템으로부터 데이터를 추출, 정제하여 데이터 웨어하우스로 적재하는 전 과정⁴⁾을 의미한다. 이 과정에는 2 가지 방식이 있다. 하나는 전통적인 코딩 기법에 의해 프로그램을 개발하는 것이고 다른 하나는 여러 운영 데이터베이스 및 외부 원천으로부터 데이터를 추출하는 기능, 이 데이터들을 정제, 변형, 통합하여 데이터 웨어하우스로 적재하는 기능, 원천에서의 갱신을 반영하고 웨어하우스로부터 데이터를 제거하여 아카이브 저장소에 넣을 수 있도록 웨어하우스를 정기적으로 재충전(refreshing) 해주는 기능을 포함하는 자동화된 도구를 사용하는 것이다. 따라서 데이터 웨어하우징 시스템은 데이터 웨어하우스로 데이터를 적재하기 위한 여러 가지의 데이터 추출 및 정제 도구와 적재 및 갱신 유틸리티를 필요로 한다(자세한 내용은 III장 참조). 외부 원천으로부터 데이터를 추출하는 것은 표준 인터페이스(예를 들어, Information Builder, EDA/SQL, ODBC, Oracle Open Connect, Sybase Enterprise Connect, Informix Enterprise Gateway 등)를 통하여 구현된다[Chaudhuri & Dayal, 1997].

(4) 메타데이터 저장소

데이터 웨어하우스에서 전사적인 업무 모델을

- 2) 자세한 내용은 Barquin, R. C., & Edelstein H. A., Planning and Designing The Data Warehouse, Prentice Hall PTR, New Jersey, 1996.의 pp. 183-187을 참고하기 바람.
- 3) 자세한 내용은 <http://members.aol.com/fmcguff/dwmodel/index.htm> 의 "Data Modeling for Data Warehouses"을 참고하기 바람.
- 4) 이 과정을 "데이터 웨어하우징(data warehousing)"이라고도 한다.

반영하고 있는 요소는 메타데이터(데이터에 대한 데이터)로, 이는 데이터 웨어하우스에 있는 여러 데이터의 다양한 측면을 기술해주는 일종의 데이터베이스이다[Abbey & Corey, 1997; Barquin & Edelstein, 1996; Devin, 1997; Inmon, 1992; Inmon et al., 1997; Poe, 1996]. 메타데이터는 데이터 모델, 일관성있는 재사용을 위한 데이터 요소의 정의, 업무 규칙, 원천 데이터베이스와 목표 데이터베이스 사이의 매핑(mapping) 관계 등을 저장하며, 최종 사용자에게 보여주는 업무 데이터 관점에서 이들을 유지한다[Barquin & Edelstein, 1996].

(5) ODS(Operational Data Store)

ODS는 매일 매일의 상세한 운영 업무의 의사 결정을 지원하는 주제 지향적이고 통합되고 휴발성이 있는(갱신이 되는) 현재의 혹은 현재에서 가까운 데이터들의 집합이다[Inmon, 1994]. ODS는 2 가지 측면에서 데이터 웨어하우스와는 다르다. 첫 번째 차이는 ODS는 휴발성의 특징이 있다면 이와 반대로 데이터 웨어하우스는 비휴발성의 특징이 있다는 점이다. 두 번째 차이는 데이터 웨어하우스는 장기간의 이력 데이터를 포함하는 반면에 ODS는 현재 또는 현재에서 가까운 시간의 데이터를 포함한다는 것이다.

(6) 데이터 마트

데이터 마트는 특별한 기능의 업무 영역(부서 또는 부문 단위의 특정 작업 집단)을 지원하기 위한 전사적인 데이터의 부분집합을 구현한 특화된 데이터 웨어하우스의 유형이다. 데이터 마트는 광고의 영향 측정, 판매량의 측정 및 예측과 같은 특정 업무와 관련된 목적을 갖는다 [Armstrong; Firestone, 1997]. 데이터 마트는 전체 데이터 웨어하우스 보다는 적은 량의 데이터를 포함하기 때문에 특정 사용자 집단에 맞는 보다 덜 복잡한 데이터 관점을 제공하고 보다

우수한 성능을 제공할 수 있다.

(7) 웨어하우스 관리 층(Warehouse Management Layer)

이 계층은 데이터 웨어하우스에 대한 접근을 관리하고 통제하는 메카니즘뿐만 아니라 성능과 활용을 감시하고 공유 데이터에 대한 일관성 있는 관점을 제공하는 관리 기능을 포함한다. 그러므로, 업무 데이터 요소의 명칭과 데이터에 대한 설명(description)을 제공하는 데이터 저장소로부터 사용자 디렉토리를 구축할 수 있는지의 여부를 고려할 필요가 있다. 또한 업무 사용자와 응용 시스템이 물리적인 자료 구조의 변화로부터 독립될 수 있도록 돕고, 데이터 안전이 보장될 수 있도록 지원하는 업무 관점을 관리하는 미들웨어의 역할을 고려할 필요가 있다[Barquin & Edelstein, 1996].

(8) 최종 사용자 접근 도구(End User Access Tools)

이 계층은 최종 사용자가 직접 다루는 층이 된다. 특히, 이 계층은 최종 사용자가 매일 일상적으로 사용하는 도구(예를 들어, Excel, ACCESS, SAS 등)들을 나타낸다. 또한, 분석과 표현을 위한 보고서, 스프레드 쉬트, 그래프, 차트를 화면에 보여주거나 인쇄하는데 관련된 하드웨어 및 소프트웨어를 포함한다. 대부분의 최종 사용자 접근 도구는 사용자 데스크탑에 상주하여 데이터를 조작하고

<표 3> 데이터 웨어하우스 활용 유형

	주요 사용자	주요 작업
질의·보고서 작성	지식 작업자	·ad hoc 질의, 상세 보고, 도표화
OLAP 및 데이터 마이닝	중간 관리자 및 분석가	·다차원 뷰, 요약/집계, 드릴 다운, 슬라이스/다이스, 통계 분석 및 인공지능 기법 이용한 지식 발견
EIS	임원	·터치 스크린 등의 이용, 다양한 자료원 이용, 브리핑, 모니터링

<표 4> 데이터 웨어하우스 관련 도구

획득 도구	
데이터 관리 도구	•데이터 저장 및 관리
데이터 모델링 도구	•모델링
데이터 추출 도구	•원천 데이터베이스에서 데이터를 꺼냄
데이터 정제 도구(scrubbing)	•불일치, 중복, 데이터 입력 오류를 교정하여 기존의 데이터를 정화
데이터 적재 도구	•정제 후 원천 데이터베이스에서 새 저장소로 레코드를 적재
무결성 검사 도구	•레코드가 원천 데이터베이스에서 새 저장소로 정확하게 적재되었는지 확인
데이터 변형 도구	•원천 데이터를 데이터 웨어하우스에 맞는 포맷으로 변환
데이터 전송 도구	•원천 데이터의 사본을 데이터 웨어하우스로 옮겨서 적절한 위치에 삽입
동기화 및 복제 도구	•전사에 걸친 데이터베이스에 트랜잭션을 복사함으로써 분산 데이터베이스를 동기화
저장소	•개발자가 프로그램내 구성요소를 추적할 수 있도록 하는 다양한 항목을 저장하는 특화된 데이터베이스
저장 도구	
데이터베이스	•관계형 데이터베이스와 다차원 데이터베이스
SMP 하드웨어	•대용량 데이터를 병렬 처리하기 위한 시스템
MPP 시스템	•대용량 데이터를 병렬 처리하기 위한 시스템
접근 도구	
대리자(agent)	•예외적 상황을 자동적으로 감지하여 보고
다차원 질의 도구	•데이터에 대한 다차원적 뷰를 제공
병렬 처리	•복잡한 질의를 할 경우 여러 프로세서에 작업을 나누어 처리하여 응답시간을 줄임
다차원 클라이언트/서버 접근도구	•전반적인 뷰를 제공하고 슬라이스/다이스 기능 제공
통계 분석	•추세 분석, 회귀 분석 등의 통계 분석 수행
데이터 발견	•대량의 원 데이터를 조사하여 새로운 유용한 정보를 발견 (discovery)
의사결정도구	•접근, 보고, 분석 기능 제공
OLAP	•온라인 분석 프로세싱 기능 제공(ROLAP, MOLAP)
데이터 시각화	•데이터를 그림 형태로 변환

분석하고 표현하는 기능을 제공한다. 초보 사용자부터 응용 시스템 사용자에게 이르기까지 모든 계층의 사용자 집단은 그들의 기술적인 숙련 정도와 복잡한 조작을 수행하기 위한 분석 요건에 따라서 기본적인 질의 및 보고를 할 수 있는 도구를 포함하여 정형화된 질의 및 다차원 분석을 할 수 있는 OLAP, MOLAP(Multi-dimensional On-Line Analytic Processing) 도구, EIS/DSS(Executive Information Systems/Decision Support Systems) 에

플리케이션, 데이터 마이닝 애플리케이션, 웹 브라우저 등을 사용한다<표 3 참조>.

III. 데이터 웨어하우스 요소 기술

3.1 요소 기술의 분류

데이터 웨어하우스 구축에 필요한 것은 크게 데이터를 기존 데이터베이스에서 뽑아오는 데이

터 추출도구와 이를 다시 원하는 형태로 가공하는 데이터 모델링 방법론, 클라이언트에서 온라인으로 데이터를 분석처리해서 볼 수 있도록 해주는 OLAP⁵⁾ 도구로 나눌 수 있다. 물론 데이터 웨어하우스에서 데이터베이스 엔진을 빼놓을 수는 없다. 이밖에 구축 이전에 사용자의 요구사항이 무엇인지 파악할 수 있도록 지원하는 요구사항 방법론 등이 필요하다. Sakaguchi & Frolick [1997]은 정기간행물에 실린 데이터 웨어하우스 문헌에 대한 실증 연구를 통해 데이터 웨어하우스에 필요한 도구를 다음 <표 4>와 같이 정리하였다. 본 절에서는 ETT 도구를 중심으로 설명하고자 한다.

3.2 ETT 도구의 기능

자료 획득 도구는 그 기능에 따라 <표 4>에서와 같이 세부적으로 분류할 수 있다. 데이터 웨어하우징 과정은 크게 추출, 정제, 적재, 재충전 과정을 거치므로 이러한 과정을 지원하려면 ETT 도구가 어떤 기능을 갖추어야만 할 것인가를 중심으로 설명하기로 한다. Squire[1995]는 운

영시스템에서 데이터 웨어하우스로 데이터를 옮길 때에 실제로 수행되는 작업은 ① 자료의 재포맷 및 재계산 ② 데이터의 키 구조 변경 ③ 데이터 웨어하우스의 데이터에 시간 요소 추가 ④ 데이터의 디폴트 값 확인 ⑤ 여러 원천 중 '최선의 데이터'를 선택하는 논리 넣기 ⑥ 데이터 요약(집계) ⑦ 데이터를 일치시킴(tallying) ⑧ 여러 원천으로부터의 데이터 병합 등이 있다고 주장하였다. 이러한 작업 과정에서 ETT 도구들이 갖추어야 할 기능들을 세부적으로 설명하면 다음과 같다.

3.2.1 데이터 획득(data capture)

자료 추출 도구는 데이터가 적재되어 있는 운영 시스템 상의 여러 원천 시스템의 다양한 데이터 구조로부터 데이터를 추출 할 수 있어야 한다. 또한, 추출 도구는 운영 시스템의 거래 프로그램(transaction programs)에서 제공되는 사용자 로그(user log)에서 제공되는 데이터뿐만 아니라 DBMS 로그에서 제공되는 데이터 또는 데이터베이스 데이터를 추출할 수 있어야 한다 [Kelly, 1994].

3.2.2 데이터 정제, 변형 및 적재(data cleansing, transformation & load)

추출 도구는 일치, 병합, 감사(audit), 변환 등의 기능을 지원 할 수 있어야 한다. 이러한 기능들은 데이터를 통합하고 변환되는 데이터의 무결성(integrity)을 유지하는데 필수적인 기능이 된다. 또한 관계형 데이터베이스로부터 다차원 데이터베이스로 데이터 변형 및 전송을 용이하고 편리하게 지원할 수 있어야 한다. 특히, 변형 과정은 목표 시스템(target system)에서 이루어지는 것이 바람직하다. 왜냐하면, 원천 시스템 환경에서 변형 과정이 발생하면 시스템에 부담을 주어 응답시간에 영향을 미치기 때문이다[Kelly, 1994]. 세부적인 기능들을 분류한 범주는 다음과 같다.

5) OLAP은 의사결정을 위한 분석작업을 온라인으로 수행하고자 하는 것이기 때문에, 과거의 관계형 모델이 2차원적인 형태, 곧 행과 열로 데이터를 저장하고, 조인과 같은 방법으로 사용자에게 데이터를 돌려주었던 것과는 다른 형태, 곧 다차원적으로 데이터를 저장하고 관리한다. 이러한 다차원 데이터베이스(MDB, Multidimensional Database)는 데이터 자체를 다차원형으로 저장하고, 데이터를 요청하는 사용자에게 사용자가 원하는 여러 차원의 데이터를 보여줄 수 있다. 다차원 데이터베이스는 관계형 데이터베이스와 비교할 때 집계 결과를 미리 계산하여 저장해 두기 때문에 속도나 데이터의 질에 있어서 상당히 앞선다고 평가되고 있다. 이러한 방식을 MOLAP(Multi-dimensional OLAP)이라고도 한다.

관계형 데이터베이스에 대해 OLAP을 수행하는 것도 가능하며 이를 ROLAP(Relational OLAP)이라고 한다. 이것은 사용자의 데스크탑이나 특정한 서버에 놓인 OLAP 도구를 통하여 관계형 데이터베이스에 저장된 데이터를 다차원적으로 보여주는 것이다. 구조는 좀 더 유연하지만 계산에 따른 부하(overhead)가 따른다. 이 두 가지 방법은 각각의 장단점이 있으며, 점차 서로 보완하는 방향으로 함께 쓰이는 추세이다[조재희, 1997].

(1) 데이터 정제

웨어하우스 내에 있는 데이터의 정확성은 매우 중요하다. 그러나 웨어하우스의 데이터는 여러 원천으로부터 많은 양의 데이터가 연관되기 때문에 데이터의 오류나 이상(anomaly)이 발생할 가능성이 높다. 그러므로 정제를 위한 도구들은 이러한 오류들을 감지하고 교정하는데 유용해야 하며 합수적 종속성과 이상을 식별하고 레거시(legacy) 파일 구조로부터 원천 데이터베이스의 구조(예를 들어, 관계형 데이터베이스)를 만들기 위하여 기존의 레거시 데이터를 자동으로 분석할 수 있어야 한다. 이러한 도구들을 사용함으로써 문서화가 제대로 이루어지지 않은 레거시 데이터 구조를 검증할 수 있으며 표준화 및 재구성을 가능하게 해준다. 또한 이러한 도구들의 일부는 데이터를 감사하고(auditing) 데이터 질을 측정하며 불필요한 데이터를 제거할 수 있도록 지원한다. 데이터의 정제가 필요한 경우의 예로는 일치하지 않는 필드 길이, 일치하는 않는 설명, 모순된 값의 할당, 입력의 누락, 무결성 제약조건의 위배 등이 있다[Chaudhuri & Dayal, 1997].

(2) 변형 및 적재

변형 및 적재 기능을 제공하는 도구는 원천 시스템으로부터 데이터를 선택하여 목표 시스템 구조로 매핑하고 변형하기 위한 프로그램 코드를 만들어줄 뿐만 아니라 목표 데이터베이스로 데이터를 한번에 적재하거나 변화가 있을 때마다 지속적인 적재를 할 수 있도록 파일을 만든다. 이러한 범주에 있는 도구들은 다수의 데이터 원천 및 파일 유형을 취급할 수 있으므로 별개의 데이터들이 웨어하우스 내로 통합될 수 있도록 지원한다. 도구 내부에서 업무 규칙들이 명시되고 일관성 있는 프로그램 코드가 만들어진다. 이들 도구들은 유도(derivation), 요약 등을 포함하여 원천 데이터로부터 목표 데이터까지 추적할 수 있는 기능을 제공할 수 있는 메타 데이터

를 만든다. 데이터의 변화가 있을 때 그 변화를 도구 내에서 명시하고 관련 프로그램이 작성되며 테스트되어 진다.

또한 적재를 위한 추가적인 처리가 요구될 수 있다. 즉 무결성 제약을 체크하거나, 정렬을 하며 웨어하우스 내에 유도 테이블(derived table)을 만들기 위하여 집계 또는 그 밖의 다른 계산을 할 수 있다. 아울러 인덱스와 또 다른 접근 경로를 만들고 목표 저장 영역을 여러 개로 분할할 수도 있어야 한다[Chaudhuri & Dayal, 1997].

3.2.3 재충전 지원(refresh support)

대부분의 추출 도구는 운영 데이터를 추출하기 위하여 수행되는 추출 프로그램을 만드는 코드 생성기에 불과하다. 그러나, 대부분의 데이터 웨어하우스 구현에 있어 데이터 갱신이 발생하는 경우, 갱신 데이터를 옮겨줄 수 있는 데이터 재충전 메카니즘을 사용하여 데이터 웨어하우스에 신속히 옮겨주는 것이 필요하다[Kelly, 1994]. 웨어하우스를 재충전하는 것은 원천 데이터의 갱신을 전파하여 이에 해당하는 웨어하우스 내의 기본 데이터와 유도 데이터를 갱신하여 주는 것이다[Mumick et al., 1997]. 이와 관련하여 두 가지의 쟁점이 있다. 즉 언제 재충전 할 것이며 어떻게 재충전할 것인가 이다. 재충전 정책은 사용자 요구, 전송 데이터 량(traffic), 원천의 특성, 데이터베이스 서버의 수용력 등을 고려하여 이루어진다.

데이터 웨어하우스는 보통 주기적으로 재충전된다. 그러나, 만약 OLAP 질의가 실시간 데이터를 필요로 한다면(예를 들어 현재 시점의 주식 거래량 등) 갱신이 이루어질 때마다 그 갱신을 옮겨 줄 필요가 있다.

재충전 방식에는 일괄 갱신에 의한 방법과 변화된 부분만을 적용하는 갱신 방법이 있다. 이 두 가지 방식에는 서로 장단점이 있다. 즉 갱신을 지원하기 위하여 일괄 적재 유틸리티를 사용

하여 제한된 시간 윈도우(보통 밤에 이루어짐) 내에서 데이터를 재충전할 수도 있으나 많은 양의 데이터를 적재하기 위하여 연속적인 적재를 하는 경우, 몇 주 혹은 몇 개월이 소요될 수도 있다는 문제점이 있다. 그러므로 원천으로부터 변화된 부분만을 추출하여 웨어하우스에 그 변화된 부분을 적용하는 것이 바람직하다. 그러나 이것은 변화된 데이터만을 골라내야 함으로 복잡한 작업 과정이 되며 정확한 참조를 하지 못하는 경우 데이터 품질의 문제를 발생시킬 수 있다. 또한 집계를 변화시키기 위하여 필요한 작업을 수행하는 것은 단순히 적재하는 것보다 더 어렵다. 따라서 상업용 유틸리티들은 웨어하우스 내로 적재되는 데이터의 양을 감소시키기 위하여 재충전하는 동안 점진적인 적재를 지원할 수 있어야 한다. 점진적인 적재는 지속적인 질의들과 충돌할 수도 있으므로 비교적 짧은 시간의 연속적인 거래를 주기적으로 처리하여 이루어진다. 예를 들어 1000 개의 레코드마다 혹은 몇 초 간격으로 이루어지는데 이러한 거래의 연속은 유도 데이터 및 인덱스를 웨어하우스 내에 있는 기본 데이터(base data)와 일관성을 보장할 수 있도록 조정되어야 한다[Chaudhuri & Dayal, 1997].

IV. ETT 도구들의 평가 및 검증

4.1 개요

본 장에서는 앞에서 제시한 데이터 웨어하우스 아키텍처 및 요소기술을 바탕으로 데이터 웨어하우스를 구축하기 위하여 요구되는 정보기술에 대한 평가항목을 제시하고, 그 평가항목에 대한 검증방법 및 검증과정에 따라 실제로 검증한 사례를 소개한다. 데이터 웨어하우스 구성요소와 관련된 정보기술 검증의 목적은 ETT와 OLAP 도구에 대한 운용환경, 기능 및 성능을 확인하며

<표 5> 정보기술 검증을 위한 대상 도구

구분	A 공급자	B 공급자	C 공급자	D 공급자
Meta Data	PRISM DSS Agent	Data Guide	IW (intelligent Warehouse)	EDA/SQL DSS Agent
ETT	PRISM	Dprop/NR(*) Dprop/R(**) Datajoiner Data Refresher	PLATINUM Source Point	EDA/SQL
OLAP	DSS Agent	Brio Query	PLATINUM	DSS Agent

*DataPropagator NonRelational

** DataPropagator Relational

메타 데이터 관리 방법을 확인함으로써 견실한 데이터 웨어하우스 구축에 대한 지침을 제시하고자 함이다.

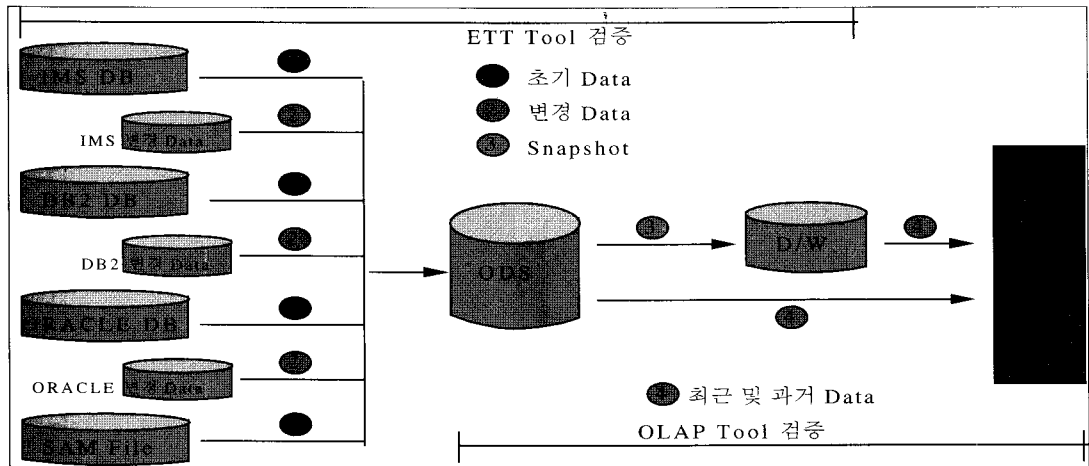
실제 검증기간은 약 2.5개월 소요되었으며, 국내 대표적인 시스템 통합(SI) 업체들이 제공하는 제품을 검증대상 도구로 선정하였다<표 5 참조>. 검증업무는 실제 K은행에서 운영중인 업무를 대상으로 하여 검증의 현실성을 제고하고자 하였다.

4.2 검증 시스템 환경

정보기술 검증을 위한 원천 데이터의 시스템 환경은 IBM 메인프레임과 유닉스로 구성되어 있고 목표 시스템 환경은 유닉스이며, 데이터 관리를 위한 CASE 툴은 Designer/2000을 사용하였다.

<표 6> 정보기술 검증 시스템 환경

	원천 시스템		목표 시스템
	IBM	Unix	Unix
H/W	IBM9672-R24	HP9000/K410	
O/S	O/S 390	HP-UX OS 10.10	
N/W	SNA , TCP/IP		
DBMS	IMS DB/DC 5.1, DB2 4.2	ORACLE 7.3.2	
FILE	SAM		



<그림 2> 정보기술 검증을 위한 데이터 웨어하우스 아키텍처

시스템 환경의 세부적인 내용은 <표 6>과 같다.

또한 정보기술 검증을 위한 데이터 웨어하우스 아키텍처는 <그림 2>와 같다. 아키텍처에서 원천 데이터베이스로는 계층형 데이터베이스인 IMS 데이터베이스에서 운영중인 K 은행의 모든 종류의 예수금 업무(보통예금, 당좌예금, 자유저축예금, 정기예금, 정기적금, 근로자정기저축 등 8개 업무 포함)와 회계업무(은행계정 전표총합계표)를 이용하였다. 회계업무의 경우 전부서 데이터를 이용하지 않고 영업부 데이터만을 적용하였다. 그리고 관계형 데이터베이스인 DB2 데이터베이스에서 운영되는 여신업무(대출업무)와 오라클 데이터베이스에서 운영되는 차입금업무를 포함하였다. 목표 데이터베이스로 ODS와 데이터 웨어하우스를 두고, ODS와 데이터 웨어하우스는 동일한 레이아웃인 관계형 데이터베이스로 구축한다. 따라서 목표 데이터베이스에는 원천 데이터베이스에 있는 예수금, 회계, 여신(대출), 차입금 데이터베이스를 만들고, 고객별/계정과목별 잔액과고객별/계정과목별 거래집계를 포함한 고객 데이터베이스를 구축한다. <그림 2>에서 ①은 원천 데이터베이스에서 목표 데이터베이스로의 초기 데이터 적재를 의미하고 ②는 원천 데이터베이스에서 거래발생에 의해 변화가 일어날 때, 순변화(delta change) 부분 데이터만

을 가지고 목표 데이터베이스를 갱신하는 과정을 설명한다. ③은 ODS로부터 데이터 웨어하우스로의 스냅샷 갱신을 의미하고 ④는 목표 데이터베이스로부터 최종 사용자들이 접근하여 데이터를 가공·분석·활용하는 과정을 나타낸다.

4.3 검증 방법

4.3.1 평가 항목 및 평가 방법

앞에서 언급한 ETT 도구들이 갖추어야 할 요건을 바탕으로 하여 다음의 몇 가지 이슈에 주안점을 두고 검증을 실시하였다. 즉 ① 여러 종류의 DBMS 및 파일 시스템으로 구성된 원천 데이터를 지원할 수 있으며 그 중에서도 특히 IBM의 DEDB (Data Entry Database)⁶⁾와 MSDB (Main Storage Database)⁷⁾을 지원할 수 있는가? ② 운영계 데이터를 추출할 경우 업무시스템에서 사용자 프로그램의 개입 없이 또는 사용자

6) DEDB는 대용량의 데이터를 신속하게 처리하기 위하여 금융권 업무에 적합하도록 설계된 IMS DB 구조로 "fast path DB"라고도 한다. 대표적인 DB 예로는 원부, 고객원장, 금융거래 내역 등이 있다.
7) 아주 빈번히 사용되는 데이터(예: 공통 코드, 고객번호 채번 등)를 Main Storage에 상주시켜 데이터 처리의 성능(performance)을 향상 시키기 위한 DB이다.

<표 7> ETT 도구의 평가 항목 및 평가 방법

평가내용		
구분	평가항목	평가방법
데이터 추출	추출 가능한 원천 DB 종류	추출가능 DB종류의 확인
	스케줄링 능력 여부	① 5분 및 ,10분 간격의 Time Scheduling에 의한 확인 ② 일제 마감 또는 특정상황의 Event Scheduling에 의한 추출 - 특정 부점 일제 마감처리후 해당 부점의 일제표 출력후 계수 확인
	원천 DB의 변화 자동 Capture	DB Archive Log 제공 여부(On-Line Log Data Set이용)
	Log File을 사용한 추출	DB Archive Log 제공 여부 (System Log Data Set 이용)
데이터 변환	여러 세그먼트 (테이블)에서 하나의 테이블로 통합이행	여신대출 거래 발생 시, 원천 테이블 '환율'에서 당일 매매 기준율을 읽어 목표 테이블 '여신대출거래실적'의 당일매매기준율 항목으로 생성
	원천시스템의 여러 항목을목 표시스템의 한 항목으로 이행	원천 테이블 '여신대출'의 적용금리구분, 변동금리구분1, 변동금리구분2의세 항목이 목표 테이블 '여신대출'의 금리구분항목으로 통합되었는지 확인
	하나의 세그먼트(테이블)에서 여러 테이블로 분할 이행	예수금거래이력 관련 원천 테이블에서 고객별과목별잔액, 고객별과목별거래집계 목표 테이블이 생성되었는지 확인
	원천의 한 항목을 목표의 여러항목으로 이행	예수금 '계좌번호' 항목의 4~5번째 문자를 계정과목으로 변환
	요약(Summarize)	고객별과목별잔액, 고객별과목별거래집계 목표 테이블의 생성
	유도 데이터(Derived Data)	목표 테이블 '저축성예금원부'의 세전만기상환액 항목이 생성되었는지 확인
	Default Value	예수금 거래Target Table에 비교1, 비교2 항목을 추가하고 각각의 Default Value 를 NULL과 'KDB'로 설정
	필터(Filter)	① 환율 테이블에 특정일자(7/1, 7/2, 7/3)를 지정하여 내용 추가 ② 대출 일일거래명세에 거래일자의 매매기준율 확인
	데이터 유형((Data Type), 길이(Length), 값(Value) 등의변환(Convert)	① 요구불예금원부, 저축성예금원부 Target Table의 실명확인여부코드, 신규여부 코드 항목의 생성확인 (Bit -> Char 변환) ② 요구불예금의 예금종류코드의 Length 조정 (C(1) -> C(3)) ③ 대출원부의 환평가여부 코드의 값 변환 (" '2'->'N' / '1'->'Y")
	외부 사용자 작성 루틴의 사용	목표 테이블 '저축성예금원부'의 세전만기상환액 항목을 User Program으로 작성 하여 확인
복잡한 Logic의 처리	회계총계정원장을 전표총합계표로부터 데이터 추출하여 가감 처리하는 루틴을 ETT 도구로 처리	
데이터 전송	원천시스템과 목표시스템 간의양방향 및 단방향 데이터 전송	환율 테이블의 Load 완료 시 'IBM 전송 자료 확인 테이블'의 해당일자 정상여부 코드 항목이 정상적으로 Update되는지 확인
웨어 하우스 관리	Performance Tuning, Monitoring 기능	대량 거래 처리 후 관리자료 출력물을 확인
	특성분석 기능	관리자료 출력물을 확인
	관리를 위한 입력내용 등록 시용이성	직접 등록에 의한 절차 확인 (구체적인 사항은 매뉴얼로 확인)
기타	오류 거래 처리	① Metadata에 등록된 내용과 상이한 강제적인 오류발생 후 처리방법: Value Check Table의 원부 관리 부점에 오류발생여부 및 재처리 방법을 확인 ② 회계 데이터 대량 변경 시 강제종료(Shutdown) 처리 후 계수를 확인 ③ 각각의 ETT 도구들의 Recovery 방법을 확인 ④ 변환 Tool 또는 Program이 원천 시스템에서 위치하여 처리하는지 혹은 목표 시스템에서 위치하여 처리하는지 확인

로그를 사용하지 않고도 추출할 수 있는가? ③ 외부정보를 내부정보로 이전하지 않고 직접 분석정보로 입력할 수 있는가? ④ 운영계에서 거래에 의해 데이터가 변경된 경우 그 변경 데이터를 즉시 또는 일정 시간 간격(예를 들어 5분 간격)으로 반영할 수 있는가? ⑤ 변환처리가 주로 목표 시스템에서 처리되는가? ⑥ 관계형 데이터베이스로부터 다차원 데이터베이스로의 데이터 이전이 쉽고 편리한가?

<표 7>에서 보여주는 바와 같이 ETT 도구 선정을 위한 평가 요소 항목을 분류하고 이를 구체적으로 검증하기 위한 방법에 따라 각각의 도구들을 비교 검토하였다.

4.3.2 검증 내용

(1) 데이터 추출

① 추출 가능한 원천 데이터베이스를 확인(<표 8> 참조)하기 위하여 IBM 시스템에서 운영되는 관계형 데이터베이스인 DB2와 계층형 데이터베이스인 IMS DEDB, MVS의 SAM 파일에서 데이터 추출이 가능한가를 확인하고, 유닉스 시스템에서 운영되는 관계형 데이터베이스인 오라클과 순차 파일에서 데이터 추출이 가능한가를 검증하였다.

② 재충전 스케줄링 가능여부는 크게 두 가지로 분류하여 검증하였다. 첫 번째 방식은 각각의 추출 도구들이 일정한 시간 간격을 정하여 주기적으로 원천 데이터로부터 데이터를 추출하여 목표시스템으로 적재되는가를 확인하기 위하여 5분 단위와 10분 단위로 각각 검증을 하였다. 예를 들어, PRISM의 경우 PSM(Prism Scheduling Manager)을 통하여 PLATINUM의 경우는 Info-Refiner를 통하여 지원이 가능한지를 확인하였으며 EDA는 EDA Copy Manager의 스케줄링 기능을 이용하였다. 두 번째 방식은 일일 거래의 마감 등 특정 이벤트가 발생하는 경우의 재충전이 가능한지를 확인하였다. 예를 들어 은행 업무의 공식적인 마감 시간은 오후 4시 30분이지만 실제로 업무를 마감하는 시간은 매일 매일 부점

이러한 검증은 각각의 추출 도구들이 일정한 시간 간격을 정하여 주기적으로 원천 데이터로부터 데이터를 추출하여 목표시스템으로 적재되는가를 확인하기 위하여 5분 단위와 10분 단위로 각각 검증을 하였다. 예를 들어, PRISM의 경우 PSM(Prism Scheduling Manager)을 통하여 PLATINUM의 경우는 Info-Refiner를 통하여 지원이 가능한지를 확인하였으며 EDA는 EDA Copy Manager의 스케줄링 기능을 이용하였다. 두 번째 방식은 일일 거래의 마감 등 특정 이벤트가 발생하는 경우의 재충전이 가능한지를 확인하였다. 예를 들어 은행 업무의 공식적인 마감 시간은 오후 4시 30분이지만 실제로 업무를 마감하는 시간은 매일 매일 부점

<표 8> 정보기술 검증 현황

① 검증 원천 DB

업무명	DB명	DB구조	내용	세그먼트명	내용	업무명	DB명	DB구조	내용	세그먼트명	내용
환율	DBAFXR	DEDB	환율	SFCMMAS	외화총계정원장	원화차입금		ORACLE		TWCISIW	원화차입승인 명부
				SFXRMAS	환율기본정보					TWCISHG	원화차입승인계획
				SFXRCUR	환율					TWCIWBU	원화차입원부
여신	DBRYBA	DB2	대출	TYDOGRS	대출거래실적					TWCWTRI	원화차입원부거래이력
				TYDOWBU	대출원부					TWCIGKW	원화차입기관코드
				TYDSIRK	대출승인이력	예수금		DEDB		SBOTDOM	보통예금
				TYDSSYN	대출승인					SYKSTIM	정기예금
		SAM		TYDOSAG	대출상환계획					SYAAMAS	일일계좌이동내역

② 검증 목표 DB

업무명	Table한글명	Table영문명	업무명	Table한글명'	Table영문명	업무명	Table한글명	Table영문명
환율	환율	TEXCYUL	여신	대출상환계획	TYDOSAG*	예수금	요구불예금원부	TYGVWBU
여신	대출거래실적	TYDOGRS	원화차입금	원화차입승인 명부	TWCISIW		저축성예금원부	TJCSWBU
	대출원부	TYDOWBU		원화차입승인계획	TWCISHG		예금거래이력	TYEKTRI
	대출승인이력	TYDSIRK		원화차입원부	TWCIWBU	기타	목표테이블 예러 내역	TTHAERR**
	대출승인	TYDSSYN		원화차입원부거래이력	TWCWTRI		IBM 전송자료확인	TIBMZSG***

* Source Data가 SAM File

** ETT 처리중 발생된 예러 내역을 보관하기

*** IBM DB2 테이블(양방향 처리 확인을 위한 테이블

위한 테이블

해환율 Target Table 생성시 사용)

에 따라 다르다. 따라서 주기적으로 스케줄링을 하는 경우 예외적인 상황에서 추출이 이루어지지 않을 수 있다. 즉 만약에 10분 간격으로 타임 스케줄링을 정했을 경우 오후 4시 30분에 스케줄링에 의해 원천으로부터 데이터 추출이 이루어진 후 어떤 부점에서 오후 4시 37분에 하루 거래의 마감("일계마감"이라 함)이 이루어졌다 가정하면 10분 간격의 시간이 아니므로 7분 사이에 발생한 원천 데이터의 변화를 반영하지 못할 수 있다. 따라서 각 추출 도구에서 일정 시간 이후(예를 들어 오후 5시 이후) 특정 부점에서 원천 데이터베이스의 특정 레코드를 변경한 경우 처리가 가능한지를 확인하였다.

③ 각 원천 데이터베이스의 로그 파일을 추출할 수 있는지를 확인하였다 이를 위하여 두 가지 방식으로 나누어 검증하였다. 첫 번째 방식은 온라인 처리 중에 원천 데이터베이스의 변화를 자동으로 감지하여 추출할 수 있는가를 확인하였다. 즉 온라인 처리 중에 변화된 부분을 DB 로그에 떨어뜨려 주면, 추출 도구가 이를 감지하여 그 로그 파일을 읽는데 문제가 없는지를 확인하였다. 두 번째 방식은 로그 데이터를 별도의 장치에 백업시켜놓은 시스템 로그 데이터를 추출할 수 있는지를 확인하였다.

(2) 데이터 변환

데이터 변환과 관련하여 검증하고자 한 항목에서 ① 여러 원천(세그먼트 혹은 테이블)으로부터 목표 시스템에서 하나의 테이블로의 통합 가능 여부 ② 원천 시스템의 여러 항목을 목표 시스템의 한 항목으로의 변환 여부 ③ 원천 시스템에 있는 하나의 세그먼트 혹은 테이블을 목표 시스템에서 여러 개의 테이블로 분할 할 수 있는지의 여부 ④ 원천 시스템의 한 항목을 목표 시스템에서 여러 항목으로 분할할 수 있는지의 여부 ⑤ 요약 기능 지원, 필터링 및 디폴트 값 처리 여부 등이 있다. 구체적인 검증 내용은 다음과 같다.

① 두 개의 세그먼트에서 하나의 테이블로 통합되는지를 검증하였다. 이를 확인하기 위하여 여신대출 거래가 발생할 경우 "환율" 원천 데이터베이스에 있는 "당일매매기준율" 항목을 읽어 들여 "대출거래실적"의 목표 테이블에 "당일매매기준율" 항목을 생성할 수 있는지를 검증하였다(<표 9>. 원천 DB 레이아웃 참고). 원천 데이터베이스가 관계형인 DB2의 대출거래실적(TYDOGRS)에 있는 "환율적용일자(FHYLYMD)"와 "통화코드(FTWACOD)" 항목을 갖고 IMS 데이터베이스의 루트와 자식의 관계로 되어있는 루트 세그먼트인 "SFXRMS(환율기본사항)"의 "SFXRURK(통화코드)" 항목과 자식 세그먼트인 "SFXRUR(환율)"의 "SFXRMSK(환율적용일자)" 항목을 매핑하여 매매 기준율 항목인 "FCBASICR"의 데이터를 추출하여 목표 테이블인 "대출거래실적"에 삽입하여 "매매기준율(FBSLYUL)"이라는 새로운 항목을 생성한다. 이와 같이 원래 없던 항목을 목표 테이블에 임의의 항목을 작성하여 만든 목표 테이블은 표 10과 같다. 원천 시스템으로부터 목표 시스템으로의 변환과정은 ETT 도구 내부에서 <그림 3>과 같이 정의된다.

<그림 3>에서 제시한 매핑 과정을 보면, 우선 Relationship Definition Header 부분에 원천 테이블과 목표 테이블을 정의하고 각각의 필드를 I/O Fields Relationship 부분에 정의한다. 여기서 목표 테이블에 추가로 선언하고자 하는 "매매기준율(FBSLYUL)"을 만들기 위하여 원천 데이터베이스인 IMS의 SFXRMS(환율기본사항)라는 세그먼트의 키 필드인 SFXRMSK(환율적용일자)를 읽어 들여 원천 데이터베이스의 대출거래실적(TYDOGRS)의 FHYLYMD(환율적용일자)와 일치하는지를 확인한다. 또한 원천 데이터베이스인 IMS의 SFXRUR(환율)이라는 세그먼트의 SFXRURK(통화코드) 항목을 읽어 들여 원천 데이터베이스의 대출거래실적(TYDOGRS)의 FTWACOD(통화코드)와 일치하는지를 확인한다.

<표 9> 원천 DB 레이아웃(일부)

원천 DB: 대출거래실적						원천 DB: 환율								
SEQ	TBL영문명	항목한글명	K	TY	LT	항목영문명	SEQ	SEG영문명	K	LV	항목영문명	항목한글명	TY	LT
1	TYDOGRS	거래번호	PK	CHAR	13	XGURBHO	1	SFXRCUR	K	3	SFXRCURK	세그먼트키통화(통화코드)	CHAR	3
:	:	:	:	:	:	:	2	:	:	:	FCGUBNCD	구분(고시, 비고시)	CHAR	1
4		원부번호		CHAR	11	XWBUBHO	:	:	:	:	:	:	:	:
5		승인번호		CHAR	10	XSYNBHO	5	:	:	3	FCCASHBY	CASH BUYING(현찰매입율)	FIXE	7.2
:	:	:	:	:	:	:	6	:	:	3	FCCASHSL	CASH SELLING(현찰매도율)	FIXE	7.2
17		원화원금		N	15	FWOHWKM	:	:	:	:	:	:	:	:
18		외화원금		N	15.3	FWIHWKM	11	:	:	3	FCBASICR	BASIC RATE(매매기준율)	FIXE	7.2
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
26		환율적용일자		CHAR	8	FHYLYMD	1	SFXRMAS	K	3	SFXRMASK	세그먼트키등록일(환율적용일자)	FIXE	8
27		통화코드		CHAR	3	FTWACOD	2	:	:	3	FCBTUSE	등록 및 공개금지 상태		
28		환율		N	7.2	FHYL	3	:	:	3	FCBTUSEI	고시통화환율등록(고시여부코드)	bit	1
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:

<표 10> 목표 DB 레이아웃(대출거래 실적)

목표 DB: 대출거래실적							설명
SEQ	TBL영문명	항목한글명	K	TY	LT	항목영문명	
1	TYDOGRS	거래번호	PK	CHAR	13	XGURBHO	
:	:	:	:	:	:	:	
4		원부번호		CHAR	11	XWBUBHO	
5		승인번호		CHAR	10	XSYNBHO	
:	:	:	:	:	:	:	
17		원화원금		N	15	FWOHWKM	
18		외화원금		N	15.3	FWIHWKM	
:	:	:	:	:	:	:	
26		환율적용일자		CHAR	8	FHYLYMD	
27		통화코드		CHAR	3	FTWACOD	
28		매매기준율		N	7.2	FCBASICR	* 생성된 항목
29		환율		N	7.2	FHYL	
:	:	:	:	:	:	:	

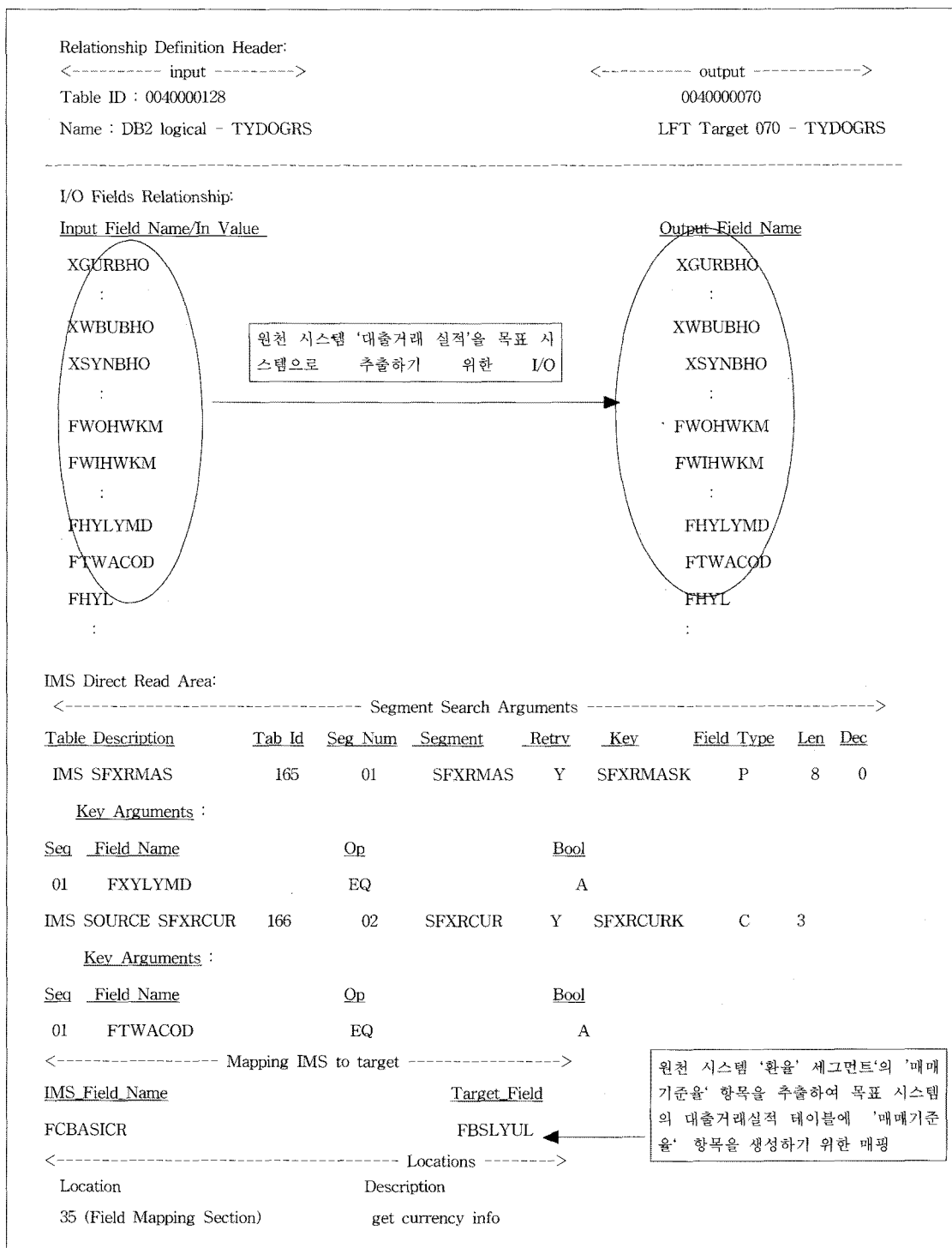
원천 시스템 '환율' 세그먼트의 '매매기준율' 항목을 추출하여 목표 시스템의 대출거래실적 테이블에 '매매기준율' 항목을 생성하기 위한 매핑

이들 2개의 항목이 일치하는 원천 데이터베이스 IMS의 FCBASICR(매매기준율)을 읽어 들여 목표 테이블에 FBSLYUL(매매기준율) 항목을 삽입한다.

② 원천 시스템의 여러 항목을 목표 시스템의 한 항목으로 변환되는지를 확인하였다. <표 11>의 "대출원부(TYDOWBU)"의 원천 테이블에 있는 항목 중 적용금리 구분(FJKLGBN), 변동금리 구분1(FBDGGB1), 변동금리구분2(FBDGGB2)의

3개 항목이 목표 테이블의 1개의 항목 즉 "금리 구분(FGRIGBN)"으로 통합되었는지를 확인하였다.

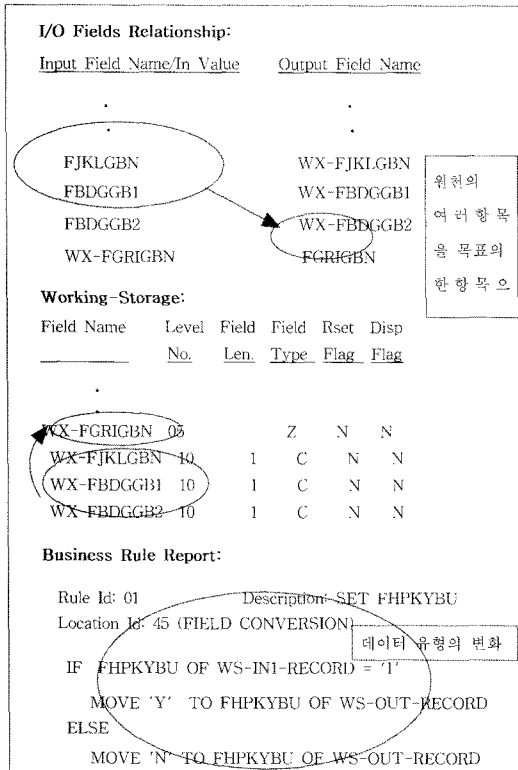
변환과정은 <그림 4>의 예제와 같이 ETT 도구 내에서 간단한 프로그램을 통하여 가능하다. I/O Fields Relationship 부분에 원천 데이터베이스의 필드와 이에 해당하는 목표 테이블의 필드를 각각 선언한다. 본 예에서는 목표 테이블의 한 항목으로 통합하고자 하는 원천 데이터 FJKLGBN, FBDGGB1, FBDGGB2를 Input Field



<그림 3> 원천 시스템의 2개 세그먼트를 목표 시스템의 1개 테이블로 통합하는 예

<표 11> “대출원부”의 원천 테이블과 목표 테이블의 매핑

원천 DB: 대출원부							목표 DB: 대출원부				
SEQ	TBL Name	K1	Field Name	Field Name	TY1	LT	Field Name	TY	LT	TBL Name	Field Name
1	TYDOWBU	PK	XWBUBHO	원부번호	A	11	원부번호	C	11	TYDOWBU	XWBUBHO
2	TYDOWBU	PK	XSYNBHO	승인번호	A	10	승인번호	C	10	TYDOWBU	XSYNBHO
3	TYDOWBU	PK	XCUSBHO	거래처번호	A	14	거래처번호	C	14	TYDOWBU	XCUSBHO
14	TYDOWBU		FHPKYBU	환평가여부	A	1	환평가여부	C	1	TYDOWBU	FHPKYBU (*)
32	TYDOWBU		FJKLGBN	적용금리구분	A	1				TYDOWBU	FGRIGBN (**)
36	TYDOWBU		FBDGGB1	변동금리구분1	A	1				TYDOWBU	
37	TYDOWBU		FBDGGB1	변동금리구분2	A	1				TYDOWBU	



<그림 4> 여러 항목을 한 항목으로 변환 및 데이터 유형 변화의 예

부분에 선언하고 이에 해당하는 임시 저장소 WX-FJKLGBN, WX-FBDGGB1, WX-FBDGGB2 를 Output Field 부분에 정의한다. 또한, 이들 3 개 항목을 통합할 임시 저장소 WX-FGRIGBN을 Input Field 부분에 선언하고 이에 해당하는 목표 테이블의 항목인 FGRIGBN을 선언한다. 하나의 항목으로 변환하기 위하여 Working-Storage 부분에서 WX-FJKLGBN, WX-FBDGGB1, WX-FBDGGB2의 상위 레벨에 WX-FGRIGBN를 정의함으로써 3개의 항목을 1 개의 항목으로 통합할 수 있다. 이렇게 통합된 항목은 OutputField의 FGRIGBN 항목과 매핑이 이루어지게 된다.

③ 데이터 유형, 데이터 값, 데이터 길이 등이 변환되는지를 확인하였다. 표 11의 “대출원부 (TYDOWBU)”의 원천 테이블에 있는 항목 중 “환평가여부 항목(FHPKYBU)”은 데이터 유형이 알파뉴메릭인 ‘1’ 혹은 ‘2’의 값 또는 널 ‘ ’ 값을 기억하고 있는데 이것의 문자형 값이 ‘N’ 혹은 ‘Y’ 가 되도록 변환한다. 변환과정은 그림 4의 비즈니스 규칙에서와 같이 ETT 내부에서 정의할 수 있다.

<표 12> 원천 시스템 '정기예금' 세그먼트

SEQ	항목한글명	K	TY	LT	LVL	항목영문명
1	계좌번호	PK	C	11	05	FMACCTNO
2	고객번호		C	14	03	FMCUSNO
3	예금종류코드		C	3	03	FMACCTCD
:	:					
20	계약기간월수		Fixed	3	03	FMTERMMM
22	월부금액		N	15		MBMKMK
:	:					
27	적용이률		Fixed	74		FMIJARAT
:	:					

<표 13> 목표 테이블 지속성 예금원부

SEQ	항목한글명	K	TY	LT	TBL	항목명	항목영문명
1	계좌번호	PK	C	11	TJCSWBU		GJWNUM
2	고객번호		C	14	TJCSWBU		CUSNUM
3	예금종류코드		C	3	TJCSWBU		YEKZRUK
:	:						
20	계약기간월수		N	3	TJCSWBU		GSGMMX
22	월부금액		N	15	TJCSWBU		MBMKMK
:	:						
27	적용이률		N	74	TJCSWBU		ZYNIYL
:	:						
35	세전만기상환액		N	15	TJCSWBU		XEZMSK

Working-Storage:

Field Name	Level	Field No.	Field Len.	Field Type	Rset Flag	Disp Flag
WX-GZNGMKC	05	4	C	N	N	
WX-FMACCTNO	05		Z	N	N	
WX-FMACCTNO-1-3	10	3	C	N	N	
WX-FMACCTNO-4-5	10	2	C	N	N	
WX-FMACCTNO-6-11	10	6	C	N	N	

분할된 항목

Business Rule Report:

Rule Id: 01 Description: SET GZNGMKC
 Location Id: 45 (FIELD CONVERSION)

```

IF WX-FMACCTNO-4-5 = '10'
    MOVE '5023' TO WX-GZNGMKC
ELSE
    IF WX-FMACCTNO-4-5 = '11'
        MOVE '5019' TO WX-GZNGMKC
    
```

예수금
"계좌번호"의
4-5번째 문
자를 "계정과
목"으로
변환하는 예

I/O Fields Relationship:

Input Field Name/In Value	Output Field Name
WX-GZNGMKC	GZNGMKC

<그림 5> 원천의 한 항목을 여러 항목으로 변환

④ 원천 시스템의 한 항목을 목표 시스템의 여러 항목으로 분할할 수 있는지를 확인하였다. 원천 시스템의 "정기예금(SJYKSTM)" 세그먼트의 항목인 "계좌번호(FMACCTNO, CHAR (11))"이 있다(<표 12> 참고). 이것을 3개의 항목으로 분할이 가능한지를 확인하기 위하여 4-5번째 문자를 분할하여 "계정과목"이라는 항목으로 변환되는지를 확인하였다. <그림 5>의 예에서와 같이 ETT 내부의 간단한 프로그램으로 지원되는지를 확인하였다. 즉 ETT 도구 내의 working-storage에 05 레벨의 WX-FMACCTNO라는 필드를 정의하고 이것을 레벨 다운(level-down)하여 3개 항목(예를 들어, 10레벨로 WX-FMACCTNO-1-3, WX-FMACCTNO-4-5, WX-FMACCTNO-6-11)으로 분할한다. 4-5번째 문자를 "계정과목" 항목으로 변환하기 위해 비즈니스 규칙 정의 부분에서 변환과정을 정의하고 이에 상응하는 목표 시스템의 항목 "GZNGMKC"을 Input/Output Field Relationship에서 정의하여 생성한다.

⑤ 원천 데이터로부터 유도 데이터(derived data) 생성 여부를 알아보기 위하여 목표 테이블(<표 13> 참조)에 "세전만기상환액(XEZMSK)"이라는 항목이 생성되었는지를 확인하였다(<그림 6> 참조).

⑥ 외부 사용자 작성 루틴을 사용할 수 있는지를 확인하고자 목표 테이블의 "세전만기상환

Working-Storage:

Field Name	Level	Field No.	Field Len.	Field Type	Field Flag	Rset Flag
WX-TAX-DAYS		05	5	N	N	N
Field Default: 0						
WX-XEZMSK		05	15	N	N	N
Field Default: 0						
WX-GSGMMX		05	3	N	N	N
Field Default: 0						

Rule ID: 04 DESCRIPTION: COMPUTE CUM TAX DAYS
 Location Id: 45 (FIELD CONVERSION)

IF PMTERMMM-RFLAG OF WS-IN1-RECORD NOT = 'ZEROS'
 PERFORM COMPUTE-04-01-IF-%1
 ELSE NEXT SENTENSE.

Derived Data 계산 루틴

Business Rule Computation Section:

PERFORM COMPUTE-04-01-IF-%1
 WX-XEZMSK = FMDUEAAMT-RFLAG *
 WX-GSGMMX + (FMDUEAAMT-RFLAG *
 WX-TAX-DAYS *
 FMIJARAT-RFLAG) / 365;

I/O Fields Relationship:

Input Field Name/In Value	Output Field Name
FMACCTNO	GJWNUM
FMCUSNO	CUSNUM
FMACCTCD	YEKZRUK
	생성된 Derived Data
WX-GSGMMX	GSGMMX
FMDUEAAMT-RFLAG	MBMKMK
FMIJARAT-RFLAG	ZYNIYL
WX-XEZMSK	XEZMSK

그림 6. Derived Data 처리 및 외부 사용자 작성 루틴의 사용

액(XEZMSK)" 항목을 사용자 프로그램으로 작성하여 확인하였다(<그림 6> 참조).

⑦ 복잡한 로직 처리를 확인하기 위해서 원천 데이터베이스로부터 데이터를 추출하여 가감 처리하는 루틴을 ETT 도구로 처리하여 보았다.

⑧ 원천으로부터 특정 레코드만을 추출할 수 있는지의 필터(filter) 기능을 확인하였다. 이를 위하여 "환율" 테이블에서 특정 일자(예를 들어 7/1, 7/2, 7/3)를 지정하여 필터 기능을 지원하는지를 확인하였다.

위의 ⑤ ⑥ ⑦의 기능을 확인하기 위하여 EDA/SQL은 Focus 언어로 구현하였고 PRISM의 경우는 COBOL로 구현하였으며 B사가 제시한 모듈에서는 C 프로그램 원천으로부터 목표 시스템으로의 SQL 로직 및 Call Procedure를 컨트롤 테이블 내에 관리하여 확인하였다. 그밖에 원천 시스템의 OLTP에 영향을 최소화하고자 목표 시스템에서 변환이 이루어지는지를 확인하였으며 원천 데이터를 요약 집계하여 목표 테이블이 생성되는지의 여부 및 디폴트 값 처리 여부 등을 검증하였다.

(3) 데이터 전송

원천 시스템과 목표 시스템 사이의 데이터 전송이 한 방향 즉 원천에서 목표 시스템으로만 이루어지는지 아니면 원천과 목표 시스템 사이에 필요에 따라 양방향으로 데이터전송이 가능한지를 확인하였다. 이를 위하여 IBM 원천 시스템에 있는 "환율" 정보를 전송하여 적재가 된 후에 IBM 전송 자료를 확인하기 위하여 테이블의 해당일자가 정상(한 방향 전송)인지 그리고 코드 항목이 정상적으로 갱신되는지의 확인 정보를 원천 시스템으로 전송(양방향 전송)하여 주는지를 확인하였다.

(4) 오류 거래 처리

① 데이터를 대량으로 거래하는 도중에 서버를 강제 종료(shutdown) 한 후 계수를 확인하고 복구와 관련된 관리 정보를 확인하였다. 즉 강제 종료 전에 원천과 목표 시스템을 확인하고 강제

종료 후에 목표 시스템을 확인하였으며 복구방안에 따라 재처리한 후에도 원천과 목표 시스템의 계수를 비교 확인하였다.

② 강제적인 오류 발생 후 각각의 ETT 도구들이 어떻게 처리하는가를 확인하였다. 우선, 오류 내용을 ETT 도구에서 관리할 수 있는지를 확인하기 위하여 목표 시스템의 “요구불예금원부” 테이블의 “계좌관리부점코드”를 오류 확인을 위한 체크 테이블로 정하여 오류 발생 시 ‘ZZZ’의 체크 값(check value)을 등록하도록 하였다. 가능하면 ETT 도구의 메터 데이터 관리 부분에서 오류 체크 값을 매핑 및 등록을 하여 데이터 오류 처리에 대한 처리 방법을 기술해줌으로써 해결 가능한지를 검증하였고 메터 데이터 관리 부분에서 지원하지 못하면 목표 데이터베이스에서 체크 값을 등록할 수 있는지를 확인하였다. 만약 ETT 도구에서 관리하지 못하면 “목표 테이블 오류 내역 테이블(TTHAERR)” 에 오류 내용을 수록하고 재처리를 할 때에 오류 내용에 대하여 “Y” Set 프로그래밍이 되는지를 확인하였다.

③ 집중처리 검증(stress test)을 위하여 “자유저축예금”, “여신대출”, “원화차입금” 등의 원천 데이터 변경 작업을 동시에 대량으로 처리하였다. 이 데이터 변경 작업을 3분 단위로 대량 처리할 수 있도록 각각의 ETT 도구에 있는 잡 스케줄러(Job Scheduler)에 등록하여 3시간 동안 집중 처리한 후 대량 거래에 의한 성능, 장애 등 계수 확인을 하였다.

④ 오류 처리를 위한 변환 도구 및 프로그램의 위치가 원천 시스템에 존재하여 이루어지는지 아니면 목표시스템에 위치하여 이루어지는지를 확인하였다.

(5) 웨어하우스 관리

각각의 도구에서 성능 조절(performance tuning), 모니터링, 특성 분석 기능 및 웨어하우스 관리를 용이하게 할 수 있도록 데이터 등의 내용 입력을 용이하게 지원하는지를 확인하였다.

4.4 검증 결과

앞에서 제시한 ETT 도구 선정을 위한 평가 요소 항목별 검증결과는 <표 14>와 같으며 그 내용은 다음과 같다.

4.4.1 데이터 추출

① 추출 가능한 원천 데이터베이스를 확인함에 있어 제안된 모든 도구가 검증 환경에서 제시된 원천 데이터베이스로부터 데이터를 추출할 수 있었다.

② 스케줄링 능력에 있어 제안된 도구 모두가 타임 스케줄링 기능을 가지고 있었다. 즉 PRISM은 PSM(Prism Schedule Manager)을 통해 스케줄 기능이 가능하였고 EDA/SQL은 EDA의 Copy Manager가 그 기능을 수행하고 있었으며 PLATINUM의 경우 InfoRefiner와 InforPump에서 지원하였다. 그러나 이벤트 스케줄링 기능에 있어서는 제안된 도구 중 일부만이 지원함을 확인하였다. 즉 PRISM만이 이 기능을 지원하고 있음

<표 14> ETT 도구의 검증결과

구분	A	B	C	D
- 데이터 추출				
·추출 작업 타임 스케줄링 기능	○	○	○	○
·추출 작업 이벤트 스케줄링 기능	○	×	×	×
·OLDS(On-line Log Data Set 이용	×	○	○	○
·SLDS 이용(System Log Data Set)	○	○	○	○
- 데이터 변환				
·여러 세그먼트에서 하나로 통합	○	×	○	○
·복잡한 로직 구현	○	×	×	○
·기타 변환 기능	○	○	○	○
·목표 시스템에서의 변환 여부	○	○	×	○
- 데이터 전송(양방향)				
- 한글 처리	○	○	○	○
- 오류거래 처리	○	×	×	×
- 복구(Recovery)	○	○	○	○
- 집중처리 테스트(Stress Test)	○	○	○	○

나 B사가 제안한 모듈의 경우 DropR Apply에서 마감 상황을 인식할 수 있는 정보(예를 들어, 부점별 마감현황)를 별도로 제공해야만 했고 나머지 도구들이 정상적으로 지원하지 못하였다.

③ 온라인 데이터 셀의 데이터베이스 로그에 대한 자동 추출 기능에 있어 PRISM은 지원하지 못하였으나 나머지 도구들은 모두 지원하였다. 그러나 시스템 로그 데이터 셀로부터의 데이터 추출은 제안된 도구 모두 지원하였다.

4.4.2 데이터 변환

① 여러 세그먼트(테이블)에서 하나의 테이블로 통합되는지의 여부 검증에 있어 B사가 제안한 도구를 제외하고 모두 지원하였다. B사가 제시한 도구는 DpropNR의 Refresher 기능에서 초기 데이터의 이행시 IMS DB, DB2, MVS의 조인 이동은 가능하나 변경된 부분의 이동은 별도의 작업이 필요하였다.

② 복잡한 로직 구현에 있어서 PRISM은 USER EXIT 기능을 이용하여 처리 가능하였고 EDA/SQL도 FOCUS 언어로 구현하였으나 나머지 도구는 충분히 지원하지 못하였다. 외부 사용자 작성 루틴 사용에 있어서는 대부분의 도구가 지원하고 있는 것으로 확인되었다.

③ 제안된 도구 모두 원천 시스템의 여러 항목을 목표 시스템의 한 항목으로 변환을 지원하였으며, 데이터 유형, 데이터 값, 데이터 길이 등의 변환도 역시 지원하였다. 그러나 하나의 세그먼트(테이블)에서 여러 테이블로 분할하거나 원천 시스템의 한 항목을 목표 시스템의 여러 항목으로 변환하는 기능, 요약 기능, 유도 데이터(derived data) 처리 기능에 있어서는 A사, B사, D사가 제안한 도구 모두 정상적으로 지원하고 있으나 C사가 제안한 도구의 경우는 Source Point라는 모듈을 추가하여 구현할 수 있었다.

④ 필터 기능에 있어서는 B사를 제외한 모든 도구들이 정상적으로 지원하였으며, 목표 시스템에서의 변환 여부에 있어서는 A사와 C사가

제시한 도구는 원천 시스템과 목표 시스템 어느 곳에서나 변환이 가능하였다. B사의 경우, Refine 부분 즉 DpropNR과 DataRefresher는 호스트에 존재하고 DpropR과 변환 프로그램은 호스트와 목표 시스템(ODS/DW)에 존재하였으며 C사가 제안한 도구의 경우에는 단순한 코드의 변환을 원천에서 구현하였고 복잡한 로직은 목표 시스템에서 Natural 프로그래밍을 통하여 구현하였다.

4.4.3 데이터 전송 및 오류 거래 처리

① 오류 거래 처리에 있어서 B사가 제안한 도구들은 목표 시스템 쪽의 오류 발생시 오류 내용을 별도로 관리하여 재처리하는 방식이 아닌 오류 부분을 해결하고 그 이후 부분을 처리하는 방식으로 제안된 도구는 이 기능을 충분히 지원하지 못하는 것으로 확인되었으며 C사와 D사가 제안한 도구의 경우도 도구 자체에서는 이 기능을 지원하지 못하고 오류 거래를 처리를 위해서는 대체 방법에 의해서만 가능함을 확인하였다.

② 그 밖의 양방향 데이터 전송, 한글 처리, 복구, 집중처리 테스트는 제안된 도구 모두 정상적으로 처리됨을 확인하였다.

IV. 결 론

데이터 웨어하우스 구성요소와 관련된 정보기술 검증의 목적은 ETT와 OLAP 도구에 대한 운용환경, 기능 및 성능을 확인하며 메타 데이터 관리 방법을 확인함으로써 견실한 데이터 웨어하우스 구축에 대한 지침을 제시하고자 함이다. 따라서 본 논문은 데이터 웨어하우스 구축을 위한 관련 정보기술에 대한 실제 검증 사례를 통하여 평가요소, 그 평가 요소와 관련된 구체적인 평가방법을 소개하고자 하였으며 이를 통하여 데이터 웨어하우스를 도입하고자 하는 기업에게 효과적인 데이터 웨어하우스 구축을 위한 관련 정보 기술 요소를 효율적으로 선택할 수 있는 지침을 제공하는데 본 논문의 의의가 있다. 데이

터 웨어하우스를 도입하고자 하는 기업이 고려해야 할 사항을 다음과 같이 제안한다.

우선 ETT 방법에 있어서 고려할 사항은 다음과 같다. ① ETT 도구를 공급하는 업체들은 일반적으로 상용화된 도구들을 제시하고 있다. 그러나 상용화된 도구들이 데이터 웨어하우스를 구축하는데 필요한 여러 가지 ETT 기능을 완벽하게 지원하지 못하는 경우가 있으므로 이러한 문제를 보완하기 위하여 공급업체들은 자체적으로 개발한 도구들을 동시에 제안하는 경우가 있으므로 상용화된 ETT 도구의 문제점과 자체 개발하여 제안한 도구들이 상용화된 도구의 문제점을 적절하게 지원하는가를 명확히 정의해야 한다. ② ETT 도구들이 모든 구조의 원천 데이터를 지원할 수 있는지를 확인하여야 한다. ③ 원천 데이터를 추출할 경우 DBMS 로그를 사용하여 추출할 수 있는지를 확인하여야 한다. ④ 원천 시스템의 데이터가 변경된 경우 그 변경 데이터를 즉시 또는 일정한 시간 간격으로 그 변화를 반영할 수 있는지를 확인하여야 한다.

둘째 ETT 검증에 있어 확인하여야 할 사항은 다음과 같다. ① 하나의 세그먼트(또는 테이블)에서 여러 테이블로 분할하거나 여러 세그먼트(또는 테이블)에서 하나의 테이블로 통합하는 경우 또는 원천 시스템의 단일 항목을 목표 시스템의 여러 항목으로 분할하거나 원천 시스템의 여러 항목을 목표 시스템의 단일 항목으로 이행이 되는지를 검증하기 위해서 필요한 항목을 적절히 선택하여 이행되는지를 확인한다. ② 유도

값(derived value), 디폴트 값 처리를 지원하는지를 확인한다. ③ 필요한 레코드만을 선택하는 필터 기능이 있는지를 확인한다. ④ 데이터 유형, 데이터 길이, 데이터 값의 변환뿐만 아니라 원천 데이터의 비트 항목을 목표 시스템에서 바이트 항목으로 변환할 수 있는지를 확인한다. ⑤ 요약 테이블 및 유도 테이블을 작성하는지를 확인한다. ⑥ 외부 사용자 작성 루틴 사용이 필요한 경우 이 루틴과의 연계 처리를 적절히 지원하는지를 확인한다. ⑦ 회계 업무와 같은 복잡한 업무를 처리하는 경우 복잡한 로직을 사용할 필요가 있는데 이러한 복잡한 로직을 ETT 도구에서 적절히 처리하는지를 확인한다. ⑧ 성능 조절(performance tuning), 모니터링, 보안, 백업 및 복구, 오류 발생 시 오류 처리 및 관리 방법 등 데이터 웨어하우스 관리 방법이 적절한지를 확인한다. ⑨ 주기적인 처리 및 이벤트에 의한 처리를 지원하는지를 확인한다. ⑩ 대량의 데이터를 처리하는 경우 성능에 문제가 없는지를 확인한다.

데이터 웨어하우스와 관련된 정보기술을 검증함에 있어 그 검증 환경에 따라 평가 방법이 다소 다를 수 있고 그에 따라 평가 결과도 차이가 날 수 있을 것이다. 본 연구에서 제시한 도구들은 데이터 웨어하우스 구축에 필요한 여러 기술 요소 중에서도 일부분을 검증하였으며 다차원 데이터베이스로의 이전에 대한 검증을 하지 못하였다는 한계점이 있다. 향후 OLAP 도구 및 메타 데이터 등의 정보기술에 대한 검증도 추가적으로 연구되어야 한다.

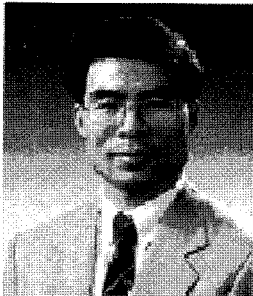
〈참 고 문 헌〉

- [1] 김기운, 서용무, 고관식, "데이터 웨어하우스 구축을 위한 데이터 웨어하우스 기술 검증에 관한 연구", *한국경영정보학회 '99 춘계 학술대회 논문집*, 1999. 6, pp. 413-422.
- [2] 박천승, "데이터 웨어하우스 환경에서의 전략적 데이터베이스 마케팅 시스템 구축", 석사논문, 고려대학교, 1997.
- [3] 조재희, "데이터 웨어하우스와 기업정보시스템", *정보과학회지*, 제15권 제 5호, 1997.5.
- [4] 조재희, "기업의 정보기반 구축을 위한 OLAP 서버", *한국경영정보학회 '95 춘계 학술대회*, 1995.6, pp. 563-577.
- [5] 조재희, "최종사용자를 위한 다차원 데이터 베이스", *한국경영정보학회 EIS/ DSS 연구분과위원회 학술대회 논문집*, 1994.12.
- [6] 홍의기, 이희석, 김태훈, "데이터 웨어하우스를 이용한 임원정보시스템 아키텍처 구현", *경영정보학연구*, 제7권 1호, 1997.6, pp. 7-24.
- [7] 홍의기, "데이터 웨어하우스를 이용한 임원 정보시스템 아키텍처 구현", *한국과학기술원 테크노경영대학원 석사논문*, 1996.
- [8] Abbey, M. and Corey, M. J, *Oracle Data Warehousing*, Osborne McGraw-Hill, California, 1997.
- [9] Armstrong, R, "Data Warehousing: Clearing the Confusion", *White Paper*, Senior Consultant NCR Corp.
- [10] Barquin, R. C., and Edelstein H. A., *Planning and Designing The Data Warehouse*, Prentice Hall PTR, New Jersey, 1996.
- [11] Chaudhuri, S., and Dayal, U., "An Overview of Data Warehousing and OLAP Technology", *SIGMOD Record*, Vol.26, No.1, March 1997.
- [12] Codd, E.F., Codd, S.B., and Salley, C. T., "Providing OLAP(On-Line Analytical Processing) to User- Analysts: An IT Mandate", *White Paper*, Arbor Software Inc., 1993.
- [13] Codd, E.F., "Twelve Rules for On Line Analytical Processing", *Computerworld*, April 13, 1995.
- [14] Devin, B., *Data Warehouse : from Architecture to Implementation*, Addison-Wesley Inc., 1997.
- [15] EDS, "Data Warehousing Primer", December 1995.
- [16] Firestone, J.M., "Data Warehouse and Data Marts: A Dynamic View", *Return to this week's issue of the Data Warehousing Career NewsLett er*, <http://www.softwareobs.com>, 1997.
- [17] Inmon, W. H., *Building the data ware house*, John Wiley & Sons, Inc., New York, 1992.
- [18] Inmon, W. H., and Hackathorn, R. D., *Using the data warehouse*, John Wiley & Sons, Inc., New York, 1994.
- [19] Inmon, W. H., Welch, J. D. and Glassey, K. L., *Managing the data warehouse*, John Wiley & Sons, Inc., New York, 1997.
- [20] Kelly, S., *Data Warehousing : The Route to Mass Customization*, John Wiley & Sons, Inc., New York, 1994.
- [21] Labio, W.J., Zhuge, Y., Wiener, J.J., Garcia-Molina, H., and Gupta, H., Widom, J., "The WHIPS Prototype for Data Warehouse Creation and Maintenance", *Processings of the ACM SIGMOD conference*, Tuscon, Arizona, May 1997.
- [22] McGuff, F. "Data Modeling for Data Ware-

- houses", [http://members.aol.com /fmcguff /dwmodel/index.htm](http://members.aol.com/fmcguff/dwmodel/index.htm), 1996.
- [23] Mumick, I. S., Quass, D., and Mumick, B. S., "Maintenance of Data Cubes and Summary Tables in a Warehouse", *Proceedings of the ACM SIGMOD conference*, Tuscon, Arizona, May 1997.
- [24] Orli, R. and Santos F., "Data Extraction, Transformation, and Migration Tools", [http://www.kismeta.com/ extract.html](http://www.kismeta.com/extract.html), 1996.
- [25] Poe, V., *Building a Data Warehouse for Decision Support*, Prentice Hall PTR, New Jersey, 1996.
- [26] Sakaguchi, T., and Frolick M.N., "A Review of the Data Warehousing Literature", *Journal of Data Warehousing*, Vol.2, No.1, January 1997.
- [27] Squire, C., "Data Extraction and Transformation for the Data Warehouse", *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pp. 446-447, 1995.
- [28] Staman, J.P., "Structuring Databases for Analysis", *IEEE Spectrum*, Oct. 1993, pp. 55-58.
- [29] Whitten, J. L., Bentley, L. D., and Barlow, V. M., *Systems Analysis and Design Methods*. (3rd ed.), Burr Ridge, IL: Irwin, 1994.
- [30] Widom, J, W.J., Zhuge, Y., Garcia- Molina, H, Hammer, J., and Labio, W.J., "The Stanford Data Warehousing Project", *IEEE Data Engineering Bulletin*, June 1995.

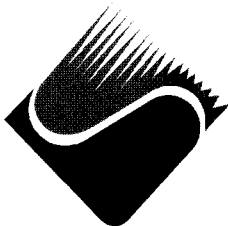
◆ 이 논문은 2000년 2월 3일 접수하여 2000년 5월 3일 게재확정 되었습니다.

◆ 저자소개 ◆



김기운 (Kim, Giun)

고려대학교 경영대학 및 경영대학원에서 강의를 하고 있고, 고려대학교 기업경영연구소에 있다. 한국외국어대학교 서양어대학영어과를 졸업하고, 경영정보대학원에서 MIS 석사학위를 취득하였으며 고려대학교 대학원에서 MIS 전공으로 박사 수료하였다. 한국산업은행에서 약 10년간 근무하면서 정보시스템 분야의 실무 경험을 하였으며, 경인여자대학 사무자동화과 및 멀티미디어정보산학부에서 전임강사로 근무하였다. 주요 관심 분야는 정보시스템 전략계획, Temporal Database, Data Warehouse, Data Mining 등이다.



서용무 (Suh, Yongmoo)

서울대학교 사범대학 수학과를 졸업한 후, 한국과학원에서 전산학을 전공하였으며, KIST 전산센터 연구원을 지낸 후, University of Texas(Austin)에서 경영정보학으로 박사학위를 취득하였다. SDS, 세종대학교, 건국대학교를 거쳐 현재 고려대학교 경영대학에 재직중이다. 주요 관심 분야는 Organizational Computing, Object-Oriented Database and paradigm, Data Mining 등이다.