제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

의미론적 계층 데이터 관리와 통합 데이터 모델을 통한 Changing Dimension 관리에 관한 연구

박경석, 김찬호, 송혜은, 유영복
한국과학기술정보연구원
e-mail : {gspark, chkim, maktup.song, yybok }@kisti.re.kr

A Study on the Changing Dimension Management Methodology With Semantic Layer Data Management and Integrated Data Model

Kyong-Seok Park, Chan-Ho Kim and Hye-Eun Song, Yong-Bok You
Korea Institute of Science and Technology Information

요 약

Business Intelligence 나 DSS 구축과 운영을 위한 근간은 기업의 통합 데이터 인프라로서의 Data Warehouse 구축이 중심을 이룬다. Data Warehouse 는 통합적, 시계열적, 비휘발적, 주제중심적 Data 로 구성된다. 이러한 특성이 이론적으로 정교함에도 불구하고 현실적인 프로젝트를 진행함에 있어서 많은 어려움을 발생시킨다. 이러한 문제의 가장 핵심적인 원인이라면 운영시스템의 변화에 따른 운영상의 리스크와 함께 Subject Area 의 요소적 변경에서 그 원인을 찾을 수 있다. 초기에 Data Warehouse 가 아무리 Business User 의 요구사항을 제대로 충족시킬 수 있다 하더라도 시간의 경과에 따라 운영시스템의 변화와 Subject Area 의 요소적 변경은 피할 수 없는 사실인데 이러한 환경에 유연하게 대처할 수 있는 Data Warehouse 가 구축되지 못한다면 결국 Data Warehouse 프로젝트는 현업의 Business 적 문제와는 거리가 먼 고비용을 투자하고 아무런 수익적 가치도 내지 못하는 거추장스러운 시스템에 지나지 않을 것이다. 본 논문에서는 Dimension 관리의 핵심이라고 할 수 있는 Changing Dimension 관리 기법과 함께 EDW(Enterprise Data Warehouse)방식의 아키텍처를 중심으로한 통합데이터모델과 함께 OLAP 메타데이터에 기반한 복합적 이면서도 현실적인 Data Warehouse 설계를 제시하고자 한다

1. 서론

기업의 데이터인프라로서의 기능과 함께 Knowledge Worker 의 지적 생산과정의 효율화를 통한 기업의 가치창출을 위해 전략적 정보시스템으로서 Data Warehouse 는 이미 가장 중요하면서도 핵심적인 위치를 차지하고 있다. Data Warehouse 는 시스템의 규모에서뿐만 아니라 그것이 차지하는 역할, 개발 및 운영의 난이도 측면에서도 상당히 높은 수준의 기술과 지식을 필요로 한다. 시스템의 성격상 Data Warehouse 는 운영시스템에 종속적일 수 밖에 없다. 즉, Data Warehouse 의 구조는 원천데이터(Source Data)를 처리, 가공하는 프로세스와 함께 이러한 운영시스템의 데이터를 중심으로 시스템이 구성되기 때문에 원칙적으로 운영시스템의 구조에 상당부분 종속되어 있다. 이러한 특성은 Data Warehouse 를 운영하는 데에 있어 엄청난 부담이 아닐 수 없다. 만약 운영시스템(OLTP)의 변화를 정확히 반영하여 Data Warehouse 를 구축하지 못할 경우 Data Warehouse 는 Business User 의 분석적 요구사항을 제대로 충족시킬 수 없으며 기업의 가치창출을 위한 전략정보시스템이라는 본연의 목적을 상실하고 만다. 결국 Data Warehouse 가 자신의 역할을 제대로

수행할 수 있기 위해서는 운영시스템의 변화와 Subject Area 의 요소적 변경에 유연하게 대처 할 수 있는 구조로 설계되어야만 한다. Data Warehouse 가 Business Intelligence 를 위한 Knowledge Worker 의 지적 생산도구로 자리매김 하고 있지만 현실에서는 잦은 시스템의 개편과 신규시스템의 도입에 따라 소스데이터의 관리에 상당한 어려움이 존재하고 이에 따라 DW 의 유지보수에 많은 어려움이 발생하게 된다. 이러한 문제를 사전에 예방하고 발생가능 한 문제를 최소화 하기 위해서는 초기 구축목적에 충실하도록 시스템의 범위를 벗어나지 않는 한도 내에서의 시스템 구축범위를 명확히 함과 동시에 운영시스템의 변경과 Subject Area 의 요소적 변경에 대한 관리를 효율적으로 수행할 수 있도록 하고 특히 Changing Dimension 에 대한 효율적 관리를 통해 Data Warehouse 데이터의 품질을 유지할 수 있는 방안이 필요하다.

2. Dimension 관리의 종류

관계형 데이터베이스를 이용하여 Data Warehouse 를 구축할 경우 다차원 모델링 과정은 필수적이다. 다차원 모델링은 Star 스키마나 Snowflake 스키마로 이원화되며 이러한 다차원 모델링의 특징은 FACT 테이블을 중심으로 Dimension 을 구성하고 이를 통해 데이터의 구조를 단순화하여 Business User 나 시스템 관리자의 데이터에 대한 이해를 높이고 한편으로 시스템의 성능을 향상시켜 OLAP Tool 등의 Access 도구를 이용하여 원활한 분석을 수행할 수 있도록 지원한다. 이처럼 다원 모델에 기반하여 구축된 데이터베이스에서 Dimension 은 운영시스템의 변화가 발생하지 않는 한 일정 기간 지속되며 FACT 의 경우는 시간의 흐름에 따라 지속적으로 증가한다. 그러나 실제 운영시스템의 변화는 어쩔 수 없이 발생하며 이로 인한 Dimension 의 변화 역시 불가피하다. 따라서 Dimension 의 변화에 대한 관리를 반드시 고려해야 하는데 Dimension 관리 기법에는 다음과 같은 3 가지 방법을 제시할 수 있다..

① 모든 것을 변경 후의 관점으로 분석
이는 새로운 값으로 기존의 Dimension 을 갱신하는 방법으로 이력관리가 되지 않기 때문에 갱신되기 전의 값이 중요하지 않거나 불필요할 경우 사용하는 방법이다.
② 변경전과 변경 후를 분리하여 관리
이 방법은 실제 Dimension 의 값과 별도로 Log Key 를 사용하여 Dimension 의 유효한 시작일자와 종료일자를 기록하고 새로운 레코드를 추가하는 방법이다.
③ 변경전과 변경후의 값을 비교할 수 있도록 관리
이러한 경우는 변화의 정도가 비교적 낮은 경우에 사용하는 방법으로 이력관리와 함께 변화하기 전의 값과 변화한 후의 값을 동시에 비교할 수 있다.

Changing Dimension 에 대한 관리는 차원테이블의 사용목적에 맞게 선택할 수 있다. 그러나 문제는 Dimension 의 관리를 얼마나 효율적으로 할 수 있느냐이다. Dimension 은 성격에 따라 갱신주기가 다양하다. 갱신이 거의 발생하지 않고 갱신되는 내용이 미미할 경우에는 Data Warehouse 관리자가 효율적으로 이를 관리할 수 있지만 갱신이 빈번하고 내용적으로도 많은 변화가 발생할 경우 이를 정확히 반영하고 관리하기에는 많은 부담이 따른다. 또한 변화의 내용이 적고 변화의 주기가 긴 경우라 하더라도 관리 프로세스가 엄격히 강요된 경우가 아니라면 이마저도 효율적으로 관리하기란 매우 힘든 일이다. 따라서 다른 부분 못지 않게 Dimension 관리를 얼마나 효율적으로 수행하느냐에 따라 데이터의 품질이 결정된다.

3. Dimension 관리의 효율화 방안

통상 Data Warehouse 의 범위와 내용을 확정짓는 운영시스템은 시스템의 전반적인 변화뿐만 Dimension 으로 분리될 수 있는 데이터의 자연스러운 변경과 함께 특정 Subject Area 내부에서의 요소적 변화는 DW 의 수정을 강요하게 된다. 따라서 운영시스템의 변화와 Subject Area 의 요소적 변경을 얼마나 효율적으로 관리하느냐의 문제는 Data Warehouse 의 데이터 품질을 결정짓는 중대한 영향요소이다. 특히 운영시스템의 변화는 코드성 정보의 변화에서 자주 발생한다. 이는 결국 Data Warehouse 의 차원정보의 변화를 의미하게 되는데 차원정보의 변화는 결국 차원과 관련된 FACT 정보에도 영향을 미치게 된다. 즉, Dimension 의 변화가 DW 전체 구조에 많은 영향을 미치게 됨을 의미한다. 그러나 이것이 운영시스템의 미미한 변화 하나하나에 까지 지나치게 민감하게 반응해야 함을 의미하지는 않는다. 즉, DW 는 주어진 Subject Area 내에서 분석적 요구를 충족시키기 위한 최초의 목적에 충실하면 되는 것이다. 그러나 적어도 Dimension 의 변화에 따른 데이터의 변화까지는 시스템이 이를 제대로 반영할 수 있어야 하는데 이를 위해서는 반드시 Dimension 관리의 효율화 과정이 선행되어야 한다. Dimension 관리의 효율화를 위해서 다음과 같은 방법을 제시하고자 한다.

(1) 지나친 역정규화 과정의 회피

통상 DW 의 FACT 테이블은 Dimension 과 집계함수로 처리할 수 있는 계수정보로 구성된 거대한 테이블이다. 이러한 FACT 테이블은 OLAP 엔진이나 기타 Query, Reporting 도구들을 이용하여 조회되는데 이때 많은 양의 데이터에 대한 집계가 이루어지고 계산이 수행된다. 따라서 이러한 목적을 위해 구성된 FACT 테이블은 OLTP 의 데이터 모델과는 차이가 있다. OLTP 에서는 Data 의 무결성(Integrity) 보장을 위해 Data 를 정규화 시켜 나가는 것이 원칙이나 DW 에서는 데이터 구조의 신속하고 정확한 이해와 처리속도 등의 이유로 인해 FACT 를 중심으로 차원정보들을 역정규화 해 나가는 방법을 택하게 된다. 그러나 특정차원의 경우 FACT 와 Dimension 의 두 가지 속성을 공유하는 정보

가 생길 수 있는데 이러한 정보는 보통 거대차원을 형성하면서 Slowly Changing Dimension 인 경우에 해당한다. 이때 처리속도 등의 향상을 이유로 역정규화 과정을 거치게 되는데 경우에 따라 이러한 정보가 처리속도에 영향을 미칠 정도로 거대하지 않고 DBMS 가 충분히 감당할 수 있다면 일정부분 정규화 시키는 것이 바람직하다. 이러한 종류의 Dimension 은 보통 의미상 계층이 존재하지 않는 Dimension 간에 물리적인 계층구조를 형성하고 있는 것이 보통이며 이들에 대한 역정규화가 적용될 경우특정 차원의 정보만 갱신됨으로 인해 전체 FACT 테이블의 구조를 변경시키지 않고도 기존의 FACT 테이블을 거의 그대로 사용할 수 있고 Dimension 을 계층구조로 모델링 할 수 있어 향후 Dimension 의 변화와 Subject Area 의 요소적 변화에 효율적으로 대처할 수 있다.

(2) OLAP 엔진을 통한 Semantic Layer 측면에서의 관리

관계형 데이터베이스를 이용한 데이터웨어하우스를 구축할 경우 ROLAP 엔진을 사용하게 되는데 ROLAP 엔진의 경우 MOLAP 엔진에 비해 매우 효율적이고 유연한 구조를 제공하고 있다. 만약 ROLAP 엔진을 이용하여 DW 를 구축할 경우 FACT 테이블과 Dimension 테이블로 구성된 Star 스키마나 Snowflake 스키마에 종속될 필요가 없다. 즉, 보통의 ROLAP 엔진이라면 관계형 데이터베이스의 구조적 측면만 Business 관점으로 재표현되는 OLAP 엔진의 의미론적 계층 수준에서만 제대로 관리한다면 구태여 복잡한 ETTL 과정을 수행할 이유가 없는 것이다. 일반적으로 대부분의 관계형 데이터베이스는 자체적으로 처리가능 한 자신만의 SQL 이나 함수 등을 지원하고 있으며 이러한 DBMS 의 특성을 ROLAP 엔진이 효율적으로 지원하고 있기 때문에 SQL 의 처리속도 문제가 심각한 수준이 아니라면 ETTL 과정과 다차원모델링을 거쳐 관리를 어렵게 할 이유가 없다.

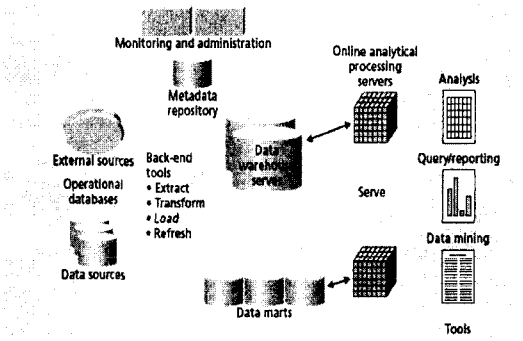

Figure 1 의미론적 계층을 통한 Business 적 재표현 구조(OLAP, Query, Reporting Tool)

그러나 이러한 유형의 데이터 관리는 DBMS 에서 제공하는 Bitmap Index 나 Summary 기능 등 DW 의 성능을 향상시킬 수 있는 여러 가지 장점을 포기해야 하는 단점도 존재함을 고려해야 한다. 이처럼 ROLAP 엔진이나 기타 이용 가능한 Access Tool 들의 의미론적 계층을 통한 관리는 Slowly Changing Dimension 에 대한 문제 뿐만 아니라 때로는 DW 시스템의 전체적인 구조를 간결하고 이해하기 쉽게 해주며 관리자로 하여금 소스시스템에 대한 이해를 보다 명확히 해주는 이점도 존재한다. DW 의 Star 스키마 구조가 이해하기 쉬운 모델인건 사실이나 다양한 Access 도구를 활용할 수 있는 상황이라면 End-User 관점에서 직접 DW 의 데이터 모델을 이해할 이유는 거의 없다. 결국 End-User 는 OLAP 엔진이나 Query, Reporting 도구 등을 이용하여 데이터를 조회하게 되는데 OLAP 에서의 Data View 는 결국 다차원 모델로 표현되고 있기 때문에 End-User 의 입장에서는 DBMS 에서 관리되는 데이터 모델이 무엇이냐에 관계없어 데이터를 조회하는데 아무런 문제가 되지 않는다. DW 를 위한 DBMS 의 데이터모델은 성능상의 목적과 OLAP 설계의 효율화가 주요 목적이지만 OLAP 설계의 효율화나 간결한 데이터 모델을 통해 얻을 수 있는 이점이 개발자나 시스템 관리자 같은 IT 전문가들에게 항상 매력적 것만은 아니다. 오히려 ETTL 과정의 복잡성이 주는 혼돈으로 인해 OLTP 의 변화에 효율적으로 대응하지 못하는 문제가 심각하게 인식되고 있기도 한 것이 현실이다.

(3) Enterprise Level 의 통합 Data Model 구현

DW 데이터의 계층적 분류는 크게 Source Data/Staging Area/ODS/DW/DM 의 순으로 분리될 수 있으며 데이터의 계층에 따른 분류를 통해 볼 때 ODS 는 통합적 성격을 유지하면서 보다 상세 수준의 데이터에 대한 분석적 요구사항에 적합한 데이터 계층에 속하면서 Data Warehouse 의 다차원 모델로 충족되지 못한 다양한 데이터를 제공하고 한편으로는 손쉬운 다차원 모델링을 지원하여 DW 의 유연한 확장을 위한 목적으로 구성된다. 그러나 보통의 경우 ODS 의 성격이 Staging Area 단계의 데이터를 크게 벗어나지 못하고 상당부분 데이터의 정규화가 이루어지지 못한 상태로 개발된다. 그러나 이러한 개발방법은 초기 개발 단계에서는 개발자의 부담을 줄여줄 수 있을 지 모르나 운영측면과 시스템의 진화 측면에서 볼 때 상당한 부작용을 초래하게 된다. 따라서 개발초기에 ODS 데이터의 엄격한 정규화는 필수적이며 이를 통해 Data Warehouse 가 내부적 환경의 변화에 유연해질 수 있다. 결국 DW 의 다차원모델을 결정짓는 원천데이터인 ODS 의 통합적이고 엄격한 정규화 과정은 DW 의 유연성 확보에 매우 중요한 역할을 담당하게 되는데 이렇게 되었을 때 DW 의 다차원 모델은 영구적 모델로서의 성격을 갖기 보다는 필요에 따라 언제든지 수정되고 새롭게 추가될 수 있는 성격의 데이터모델로 분류될 수 있다. 이는 정규화된 ODS 를 통해 언제든지 새로운 주제분야의 발굴과 함께 원시데이터에 대한 ETTL 과정이 손쉽게 수행될 수 있기 때문에 가능한 것이다. ODS 는 DW 를 구성하기 위한 운영시스템의 통합 데이터 모델로서의 역할을 담당한다. 따라서 실

제 DW 의 다차원 모델에 비하여 난이도가 훨씬 높고 데이터의 품질에 중대한 영향을 미치게 된다. Data Warehouse 의 기본 개념은 주제중심적, 통합적, 역사적, 비휘발성 데이터의 총체이다. ODS 가 Data Warehouse 의 구성 Component 로서 Data Warehouse 의 기본 개념에 비추어 볼 때 주제중심적이라는 개념에는 충분하지 못하다. 그러나 운영시스템의 데이터모델과 비슷한 형식을 유지하며 Data Warehouse 의 기본속성을 충족시키고 있기 때문에 다차원 모델에 비해서 운영시스템의 변화에 보다 유연하게 대처할 수 있으며 내부적 환경변화에도 효율적으로 대응할 수 있다. 이러한 이유로 통합적 데이터 모델로서의 성격을 갖는 ODS 의 엄격하고 정교한 설계에 많은 시간과 노력을 집중해야 할 필요가 있다.

4. 결론 및 기대효과

Data Warehouse 는 기업의 데이터 인프라로서의 단순한 개념을 뛰어넘어 Knowledge Worker 의 업무를 지원하고 기업 의사결정의 근간을 담당하고 있는 전략적 정보시스템으로서의 역할을 수행하는 중요한 위치를 차지하고 있다. 그러나 Data Warehouse 가 본연의 목적을 달성할 수 있기 위해서는 현업담당자들의 분석 요구사항을 적시에 충족시킬 수 있어야 함에도 불구하고 운영시스템의 변화와 분석을 위한 요소적 변화에 종속적일 수 밖에 없는 Data Warehouse 의 특성상 많은 위험부담을 프로젝트 초기부터 안고 있다. 이러한 문제는 주제중심적(Subject Oriented), 역사적(Time variant), 통합적(Integrated), 비휘발성(Nonvolatile) 데이터의 총체라는 Data Warehouse 의 기본 개념에 사로잡혀 이러한 문제의 해결에 적합한 데이터 모델로 제시되는 다차원모델로 구성된 시스템 설계에만 집중하여 해결하려 하는 데에서 그 원인을 찾을 수 있다. 비록 다차원 모델이 데이터에 대한 이해를 높이고 대량의 데이터를 처리해야 하는 DW 의 특성상 성능향상을 위해 적합한 데이터모델임에는 틀림없는 사실이지만 Data Warehouse 시스템의 목적을 충족시키기에는 다차원 모델만으로는 부족하다. 이러한 문제는 시스템의 유연성 확보측면에서 두드러지게 나타나는데 이를 해결하기 위해서는 다계층의 데이터 모델의 제시가 필요하며 운영시스템에 유연하게 대처할 수 있는 통합적 데이터모델로서의 ODS(Operational Data Store) 설계는 필수적인 요소이다. 이는 ODS 데이터 모델의 정규화와 함께 Dimension 관리의 효율화를 통한 원천데이터의 품질관리를 통해 가능하다. 또한 지나치게 잦은 변경이 발생하는 OLTP 의 경우 다차원모델링을 위한 데이터베이스(DW)와 ETTL 프로세스의 수정을 통하여 문제해결을 하기 보다는 OLAP 이나 기타 End-User 측면의 Access Tool 에 의한 의미론적 계층 관점에서의 다차원 모델링만을 지원하여 End-User 의 Business 적 목적 달성에 무리를 주지 않으면서도 시스템 개발자와 관리자의 변화관리에 대한 부담을 충분히 줄여줄 수 있다. 따라서 이러한 원칙에 기초하여 Data Warehouse 를 개발하게 되면 크건 작건 잦은 운영시스템의 변화와 내부적 환경변화에 필요이상으로 민감하게 반응하지 않아도 되며 특히 Dimension 의 변화와 Subject Area 의 요소변경에도 쉽게 대처할 수 있다. 이를 통해 데이터의 품질향상과 함께 관리의 효율성을 증대시켜 Knowledge Work 의 지적 생산능력 향상과 기업의 의사결정지원 프로세스의 효율화를 위해 구축된 Data Warehouse 의 본래 목적을 보다 쉽게 달성할 수 있을 것이다.

참고문헌

[1] Christopher Adamson / Michael Venerable "Data Warehouse Design Solution". John Wiley & Sons

[2] Gill, Harjinder / Prakash Rao "Client/Server Computing Guide to Data Warehousing". Que Corporation

[3] Inmon, W.H / Claudia Imhoff /Greg Battas "Building the Operational Data Store" John Wiley & Sons

[4] Inmon, W.H "Building the Data Warehouse", 2nd ED. QED Publishing Group

[5] Kimball, Ralph / Laura Reeves / Margy Ross / Warren "The Data Warehouse Lifecycle Toolkit : Tools and Techniques for Designing, Developing, and Deploying Data Marts and Data Warehouses". John Wiley & Sons

[6] Kimball, Ralph "The Data Warehouse Toolkit". John Wiley & Sons

[7] Poe, VidETTLe "Building a Data warehouse for Decision Support". Prentice Hall

[8] Silverston, Len / Inmon, W.H / John Zachman / Jonathon Geiger "Data Stores, Data Warehousing, and The Zachman Framework". McGraw-Hill

[9] Lambert, B. Data Warehousing Fundamentals "What You Need to Know to Succeed. Data Management Review, March 1996"

[10] O'Rourke, Carol/ Fishman, Neal/ Selkow, Warren "Enterprise Architecture Using the Zachman Framework". Course Technology Ptr

[11] Michael J. Corey, Michael Abbey " Oracle Data Warehousing". McGraw-Hill

[12] JF Francois Weilbach Herna L Viktor "A Data Warehouse for Policy Making: A Case Study". IEEE 1999

[13] M. Hernandez and S. Stolfo, "The Merge/Purge Problem for Large Databases," Proc. SIGMOD Conf., ACM Press, New York, 1995

[14] K. Hahn, C. Sapia, and M. Blaschka, "Automatically Generating OLAP Schemata from Conceptual Graphical Models," Proc. ACM 3rd Int'l Workshop Data Warehousing and OLAP, ACM Press, New York, 2000