데이터웨어하우스 구축을 위한 메타프로세스의 표현방법

안연식* 이춘열** 이국철**

A Metaprocess Representation Scheme for Data Warehouses

Ⅰ. 개요

데이터웨어하우스를 구축하기 위해서는 데이터 추출, 데이터 변환, 데이터 전송 등의 3단계 처리과정을 거친다[3][4]. 여기서 데이터 추출(Data Extraction)은 데이터웨어하우스의 원시데이터로서 운영데이터베이스에 저장된 정보를 식별하고 이에 접근하는 과정이다. 데이터 변환(Data Transformation)은 원시데이터를 더욱 의미있는 정보로 변환하는 과정인데, 예를 들면 집계(aggregation) 또는 표준화(standardization) 등이 포함된다. 그리고 데이터 전송(Data Propagation)은 변환된 데이터를 데이터웨어하우스에 물리적으로 적재하는 과정으로서 예를 들면 UNIX 서버에서 추출하여 변환된 데이터를 AS/400 컴퓨터의 데이터웨어하우스에 이동시키는 과정이다.

이와 같은 데이터웨어하우스 구축과정에서 변환프로세스의 목적은 첫째, 데이터웨어하우스 내에서 데이터의 품질을 향상시키고 둘째는, 데이터의 활용성(usability)을 높이기 위함이다. 따라서 고품질의 데이터로서 사용자에게 유용한 정보를 제공할 수 있는 데이터웨어하우스를 구축하기 위해서는 다양한 데이터 변환기능을 사용자가 쉽게 활용할 수 있는 데이터웨어하우스 구축도구가 필요하다. 또한 데이터웨어하우스에 저장된 데이터를 활용하는 사용자들에게 데이터의 상세한 정보를 설명해주기 위한 메타정보의 역할이 중요하다.

Ⅱ. 데이터웨어하우스와 메타정보

메타정보 또는 메타데이터는 데이터에 대한 정보를 말한다. 데이터웨어하우스 구축도구에서 메타데이터는 데이터웨어하우스 관리자에 의해 사용되는 기술적 데이터와 데이터웨어하우스 사용자에 의해 사용되는 비지니스 데이터로 구분된다[5]. 여기서 기술적 데이터는 원시데이터와 데이터웨어하우스의 구성을 설명하는 데이터이며, 비지니스 데이터는 데이터웨어하우스에 적재된 시점이나 신뢰성 등을 비지니스 관점에서 설명한다.

이들 기술적 또는 비즈니스 메타데이터들 중, 사용자들의 일차적인 관심은 데이터의 변환과정에 대한 정보라고 할 수 있다. 즉, 데이터의 형식, 길이, 값의 범위 등과 같은 데이터 자체에 대한 정보들 보다는 데이터웨어하우스에 적재된 자료가 어떤 변환과정을 거쳐 생성된 자료인지를 알려주는 변환프로세스에 관한 정보가 더욱 중요하다고 할 수 있다[7]. 그러나 대부분의 데이터웨어하우스 에서 구현되고 있는 메타데이터들은 데이터항목의 속성정보를 위주로 한 것이며, 변환프로세스에 관련된 기록 등은 상세히 관리되지 않고 있다[4][8]~[12].

변환프로세스에 대한 정보는 데이터의 변환에 대한 정보라는 점에서 데이터의 구조나 형식을 위주로 설명하는 메타데이터와는 차이가 있다[9]. 따라서 본 연구에서는 이를 기존의 메타데이터와 차별화하기 위해 메타프로세스 정보라고 칭한다.

Ⅲ. 메타프로세스 모형

메타프로세스 정보는 데이터웨어하우스 구축과정의 데이터변환 프로세스로부터 도출된다. 따라서 본 연구에서는 데이터 변환 프로세스의 유형을 정형화하고, 이를 이용하여 메타 프로세스모형의 구조를 설계한다.

데이터변환 프로세스를 모형화하면 개체관

* 경원전문대학 비서학과

** 국민대학교 정보관리학부

계(E-R)도로 표시될 수 있으며, 메타프로세스 정보는 이 개체관계도로부터 변환개체, 변환 조건 및 변환 프로세스를 중심으로 정형화된다.

1. 변환 개체의 정형화

변환개체는 주로 목표테이블을 구성하는 요소인 필드와 레코드로서 데이터웨어하우스 사용자가 메타프로세스 정보를 질의하는 대상개체가 되며, 변환프로세스를 설명하는 릴레이션에서 키항목인 "변환개체_id"필드를 말한다. 이들의 표현방법은 변환개체가 필드(즉, 테이블의 컬럼)인 경우에는 (데이타베이스명).테이블명.필드명"이 되며, 변환개체가 테이블인 경우는 (데이타베이스명).테이블명이 된다.

변환개체가 레코드인 경우는 이들 레코드들을 지정하기 위한 변환 조건이 다음 절에서 명시되는 바와 같이 정형화된다. 그러나 만약 테이블을 구성하는 모든 레코드들이 해당될 경우에는 조건이 지정되지 않는다.

2.변환조건의 정형화

변환조건은 단일조건과 복합조건 및 무조건으로 유형화될 수 있다.

① 단일조건의 경우: 예를 들면 "국가명이 대한민국이면"과 같이 1개의 조건으로 구성된다.
② 복합조건의 경우: 예를 들면 "국가명이 코리아 또는 대한민국이면"과 같이 2개 이상의 조건으로 구성된다.
③ 무조건의 경우: 예를 들면 "모든 국가명에 대하여 "의 조건은 아무런 변환 조건이 명시되지 않는다.

변환조건릴레이션은 원칙적으로 사용자가 데이터웨어하우스 구축을 위해 지정한 레코드 추출 조건이다. 따라서 위의 예에서 보다 복잡한 변환조건도 표현될 수 있으며, 데이터 추출도구에서 생성한 SQL문으로부터 채워진다.

변환조건릴레이션(변환조건_id, SQL문)

3. 변환프로세스의 정형화

변환프로세스는 그 유형에 따라서 여러 가지 관점에 따라서 정형화될 수 있다. 본 연구에서는, 첫째, 원시데이터개체에 가해지는 조작유형에 따라서, 둘째, 변환프로세스의 다중성에 따라서 정형화하였다.

3.1 원시데이터개체에 가해지는 조작유형에 따른 정형화

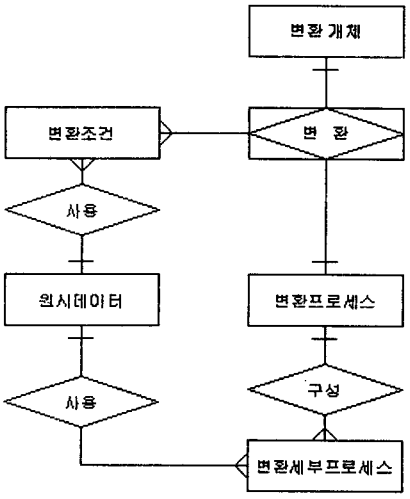
변환프로세스는 원시데이터개체에 가해지는 조작유형에 따라서 첫째, 원시데이터를 변환하여 데이터웨어하우스의 항목을 생성하는 개체변환프로세스와 둘째, 원시데이터의 항목에는 없는 항목을 새로이 생성하는 개체생성프로세스로 나뉘어진다. 개체변환프로세스는 다시, 첫째로 원시데이터개체의 속성은 계승되며 값의 갱신을 통한 변환개체를 생성하는 유형과, 둘째로 원시데이터개체의 유형(type) 또는 항목명을 변경하는 유형으로 나눌 수 있다. 그리고, 개체생성프로세스는 새로운 필드를 생성하는 경우와 레코드를 생성하는 유형으로 분류된다.

3.2 변환프로세스의 다중성

개체변환프로세스 또는 개체생성프로세스를 통해서 원시데이터개체가 목표테이블에 새로운 개체로 변환될 때 어떤 원시데이터개체는 하나의 프로세스 즉 단일변환프로세스로서 변환이 완료되는 경우도 있으나, 개체에 따라서는 일련의 변환프로세스의 수행, 즉 다중변환프로세스를 통해서 최종적인 변환개체로 생성되는 경우도 있다. 단일변환프로세스를 거친 변환개체와 달리 이러한 다중변환프로세스까지를 메타프로세스정보로 확보하기 위해서 각 변환개체(변환개체_id)에 가해진 여러 개의 프로세스를 '프로세스_id'에 추가하여 일련의 변환순서(프로세스_seq_no)에 따라 확인할 수 있다.

3.3 변환관계의 정형화

변환관계는 원시데이터로부터 변환개체가



<그림 2.a> 변환프로세스의
개체관계(E-R)도

생성되는 관계를 나타낸다. <그림 2.a>에서 예시된 바와 같이 변환관계는 원시 데이터, 변환 개체, 변환프로세스 및 변환 조건 사이의 관계릴레이션으로 표현된다. 그러나 변환 프로세스는, 변환 프로세스의 다중성에서 살펴본 바와 같이, 세부 변환 과정으로 분해되며, 원시데이터 개체로부터의 변환 프로세스는 이들 세부 변환 과정에 의하여 설명된다.

변환관계릴레이션(*변환개체id*, 변환 조건_id, 변환 프로세스_id)

3.4 세부 변환 프로세스의 정형화

원시데이터로부터 변환개체의 생성은 세부 변환 프로세스에 의하여 구체화된다. 이들 각각의 프로세스에 대해 릴레이션 모형을 설계하면 다음과 같다.

(1)필드값 변환 프로세스; 이 프로세스는 원시테이블의 레코드 또는 여러 레코드로부터 일정한 필드의 값을 변환하여 데이터웨어하우스에 저장하되 원시데이터의 속성이 목표테이블에 그대로 유지되는 프로세스이다. 예를 들면 특정 필드값을 정해진 변환식에 의해 대체(replace), 틀린 값의 정정(update), 순위 계산(scoring), default값 부여(default) 등이 여기에 속한다. 필드값 변환에 대한 메타 프로세스는 다음의 릴레이션으로 표현된다.

필드값변환프로세스릴레이션(*프로세스_id*, *프로세스_seq_no*, 원시데이터개체_id, 필드값 변환 프로세스 유형, 변환식 또는 변환참조용 테이블명, 사용자설명, 시스템설명)

여기서 *프로세스_id*와 *프로세스_seq_no*는 세부 변환 프로세스를 식별하는 키이다. 원시데이터개체_id는 원시 데이타(source column)를 나타내며, 변환 개체_id와 같은 방법으로 부여된다. 변환 프로세스 유형에는 sum, converse 등 프로세스 유형들이 저장된다. 또한 변환식에는 변환처리에 사용된 공식이 저장되거나, 변환 참조용 테이블 명이 저장된다. 또한 2개의 설명항목에는 사용자, 시스템에서 사용하는 설명정보가 각각 저장된다.

(2) 필드속성변환 프로세스; 이 프로세스는 필드의 속성(길이, 유형, unique허용유무 등)이 변경되어 목표테이블의 필드를 형성하는 프로세스로서 예를 들면 속성의 재정의(redefine) 또는 변경(reformat), 필드명 변경(rename), 별명의 변경(alias) 등이 여기에 속한다. 필드 속성 변환 프로세스릴레이션은 다음과 같이 정의된다.

필드속성변환프로세스릴레이션(*프로세스_id*, *프로세스_seq_no*, 원시데이터개체_id, 필드 속성변환 프로세스 유형, 변경전 속성, 변경후 속성, 사용자설명, 시스템설명)

(3)필드생성 프로세스; 이 프로세스는 목표테이블에 새로운 필드를 생성하여 추가하는 프로세스이다. 따라서 원시테이블에 속한 개별 필드의 값을 합산(sum), 복수 개 필드의 결합을 통한 필드생성(merge), 원시테이블의 한 필드를 복수개의 필드로 분리하여 생성되는 필드생성(split), 여러 개 필드에서 부분적인 값의 연결을 통한 새로운 필드의 생성(concatenate) 등이 이에 속한다. 필드 생성 릴레이션은 다음과 같이 정의된다.

필드생성프로세스릴레이션(*프로세스_id*, *프로세스_seq_no*, 원시데이터개체_id, 필드 생성 프로세스 유형, 변환식 또는 변환참조용테이블명, 사용자설명, 시스템설명)

(4)테이블 생성 프로세스: 이 프로세스는 원시데이터로부터 추출된 하나 이상의 필드를 이용하여 목표테이블의 레코드를 생성하는 프로세스이다. 레코드는 하나 이상의 필드로 구성되므로 레코드 또는 테이블 생성 프로세스는 필드갯수만큼의 필드생성 프로세스를 수반한다. 또한 조건에 맞는 레코드를 추출(select), 특정 필드가 조건에 맞지 않는 이유로 일어난 레코드의 탈락(filter), 복수개의 레코드 결합(householding) 프로세스 등이 여기에 속한다. 다음과 같은 릴레이션으로 정의된다.

테이블생성프로세스릴레이션(*프로세스_id*, *프로세스_seq_no*, 원시데이터개체_id, 테이블생성 프로세스 유형, 변환식 또는 변환참조용 테이블명, 사용자설명, 시스템설명)

IV. 결론 및 향후 과제

본 연구에서는 데이터웨어하우스를 구성하는 데이터에 대해서 사용자들에게 원시데이터로부터 어떤 변환과정을 거쳐 생성된 것인지를 설명해주는 메타정보의 표현방법으로서 메타프로세스 모형을 제안하였다. 이를 위해 데이터변환프로세스를 분석하여 변환조건, 변환프로세스 그리고 변환개체를 정형화하였다. 특히 변환프로세스는 원시데이터에 가해지는 조작유형에 따라서 4개 유형으로 정형화되었으며, 변환프로세스의 다중성과 변환용 참조테이블의 사용유무에 따라서도 정형화되었다. 이 모형은 변환프로세스 정보를 설명식(descriptive)으로 관리하는 것이 아니라 관계형 릴레이션으로 구현되어 있기 때문에 메타프로세스 릴레이션을 이용한 연산 및 조작이 가능하여 그 자체로서의 확장성과 유지보수성이 높다 하겠다.

향후 연구되어야 할 과제로는 메타프로세스 정보 저장소의 구현과 이의 유용성에 대한 실증적 검증 등이 있다.

참고문헌

[1] Bischoff, Joice and Ted Alexander, *"Data Warehouse from the Experts*: Chap. 13, Data Transformation," Prentice-hall: NJ, 1997, pp. 160-173

[2] Steinacher, S., *"Data Warehousing and the AS/400,"* Duke Press, 1998

[3] Inmon, W. H., *"Building the Data Warehouse(2nd Ed.)*: Chap. 3, The Data Warehouse and Design," Wiley, 1996, pp. 73-144

[4] Devlin, B., *"Data Warehouse from Architecture to Implementation : Chap. 6, Principles of Data Warehousing Design Techniques,"* Addison-wesley, 1997, pp. 108-122

[5] IBM, *"Data Warehousing Concepts for AS/400,"* IBM, 1995, pp. 5-14

[6] Flanagan, Tom, Elias Safdie, *"Meeting the Data Integration Challenge, "* http://www.techguide.com/dw/sec/dataint.pdf

[7] Flanagan, Tom, Elias Safdie, *"Putting Metadata to Work in the Warehouse,* "http://www.techguide.com/dw/sec/matadat.pdf

[8] Burch, George, *"Will the Real Metadata Please Stand Out?, "* http://www.datawarehouse.com/resource/articles/burch7.htm

[9] Cleary, John, G. Homes, S. J. Cunningham, I. H. Witten, *"Metadata for Database Mining,"* The Proceedings of the IEEE Conference on Metadata; Silverspring: Maryland, 96. 4

[10] Jordan, Arthur, *"Warehouse Data Integrity: How Long and Bumpy the Road,"* http://www.data-warehouse.com/resource/articles/jord8.htm

[11] Matadata Coalition, *"Matadata Interchange Specification(MDIS) Version 1.1,"* Matadata Coalition, 1997. 8, http://www.he.net/~metadata

[12] Inmon, W. H, C. Imhoff, R. Sousa, *"Corporate Information Factory; The Metadata Component,"* Wiley, 1998, pp. 161-162