2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

시스템 통합을 위한 코드 일관성 검사

임재걸∪, 이영재
동 국 대 학 교 컴퓨터학과, 정보관리학과
yim@wonhyo.dongguk.ac.kr, yjlee@mail.dgu.ac.kr

A Code Consistency Test in System Integration

Jaegeol Yim∪, Yeungjae Lee
Dept. of Computer Science, Dept. of Information Management, Dongguk University

요 약

시스템 통합은 개별적으로 운영되는 여러 시스템을 하나의 시스템으로 모두 합치는 것을 의미한다. 시스템 통합을 통하여 데이터를 치밀하게 조직하고, 동일한 기능의 여러 함수들을 통합하는 등의 효과를 얻을 수 있다. 시스템 통합과정에서 필연적으로 수행해야 할 작업 중 하나가 각 시스템별로 사용하던 데이터베이스들을 모두 합하여 하나의 통합 데이터베이스를 구축하는 것이다. 데이터베이스 통합 과정에서 수행해야 하는 일 중 하나가 각 데이터베이스에서 사용하는 코드시스템을 통합하는 것이다. 본 논문에서는 기존의 여러 코드 체계들을 통합하는 작업의 모델을 구축하고, 통합 코드 체계를 작성하는 작업 환경을 제공하는 소프트웨어를 설계한다.

1. 서론

CALS는 제품업체의 제조 공정과, 운용 지원 과정에서 사용하는 데이터를 표준화된 형식으로 컴퓨터에 의해 통합/자동화하는 것이다. EDI는 기업, 공공기관, 학교 및 연구소 등에서 정형화된 거래 서식을 표준 포맷과 코드 체계를 사용하여 상호 컴퓨터간의 직접 통신에 의해 교환하는 자료처리 방식을 의미한다[1].

따라서, CALS나 EDI를 실현하려면 기존의 DB를 통합하여야 하며, 이때 각 시스템에서 사용하는 코드 체계들을 하나로 통합하는 작업도 수행해야 한다. 기존의 코드 체계들을 하나로 통합하려면 코드들 간의 관계를 파악하여 이름은 다르지만 내용이 같은 코드는 하나로 통합하고, 이름은 같지만 내용이 다른 코드는 다른 이름을 부여하며, 코드간에 어떤 관계가 성립하면 이러한 관계를 명백하게 명시하여 줌으로써 데이터베이스 관리에 사용할 수 있도록 도움을 주어야 한다.

2. 관련연구

코드란 컴퓨터로 처리할 때 데이터의 분류 및 조합을 용이하게 수행하기 위해서 사용되는 기호로써 일정한 규칙에 따라 작성된다. 기호로서 코드의 적합여부는 정보처리 효율과 처리된 정보의 이용 가치에 많은 영향을 미치게 된다.

코드를 설계할 때는 코드의 경제성, 유연성, 적합성, 편리성을 고려해야 하는데, 경제성이란 가능한 한 적은 자릿수로 많은 정보를 표현하라는 것이고, 유연성이란 정보 시스템 개발 현장에서 새로 발생하는 데이터를 효과적으로 코드화시킬 수 있도록 코드 시스템을 설계하라는 것이며, 적합성은 컴퓨터로 처리하기 쉽게 코드 체계를 설계하라는 것이다. 코드의 편리성이란 사용자가 사용하기 편리해야 한다는 것이다.

코드의 종류에는 다음과 같은 것이 있다. 코드화 대상을 발생 순서나 크기 등의 순서대로 일련번호를 부여하는 순서코드가 있다. 블록 순서 코드란 코드화할 대상들을 임의의 특징을 기준으로 집단으로 나눈 다음, 집단 내에서는 순서코드를 적용하는 방법이다. 그룹분류코드는 대상 항목에 대한 분류 기준에 따라 대분류, 중분류, 소분류로 나누고, 각 분류 안에서 번호를 순서적으로 배당하는 방식이다. 이 방식은 각 자리가 특정한 의미를 가지고 있어서 분류 기능이 탁월하다. 10진 분류 코드는 그룹 분류 코드를 응용한 것으로 코드화 대상 항목을 분류하고 번호를 부여하는 방식은 동일하지만 10진수만을 코드 번호로 사용하므로 각 분류마다 오직 한 자리만이 허용되는 방식이다. 끝자리 분류 코드는 코드의 끝에 한 자리를 추가하여 항목을 분류할 수 있도록

지원하는 코드다. 연상코드는 코드 값으로 대상을 쉽게 연상할 수 있는 코드다. 유효 숫자 코드는 항목에 관련된 물리적인 수치를 코드의 일부로 사용하는 방식이다. 합성 코드는 두 개 이상의 코드를 조합하여 만든 코드 방식이다.

3. 통합과정 모델 구축

기아자동차가 현대자동차와 합병되었다고 하자. 초기에는 기아자동차와 현대자동차의 기존 시스템을 그대로 분리하여 사용할 수 있겠지만, 종국에는 이들을 통합하여 하나의 시스템으로 통합할 것이다. 이러한 현상은 기업간의 통합에서만 발생하는 것이 아니다. 같은 기업 내에서도 부서별로 필요한 시스템을 독자적으로 개발하여 운영하다가 이들을 통합하는 사례는 얼마든지 있다.

이러한 경우에 각 시스템에서 사용하는 데이터베이스를 통합하는 작업도 필연적으로 이루어져야 하고, 데이터베이스를 통합하려면 데이터베이스에 사용되는 코드체계를 통합하는 작업도 필연적으로 수행되어야 한다. 본 논문에서는 코드체계 통합 작업의 일반적인 모델을 제시하고, 코드체계 통합 작업을 원활히 수행할 수 있는 환경을 제공하는 소프트웨어를 설계한다.

데이터베이스를 통합하려면 각 시스템에서 사용하는 테이블을 비교하여 동일한 테이블은 하나로 통합한다. 속성도 마찬가지로 동일한 속성은 하나로 통합한다. 본 논문에서 다루고자 하는 코드의 통합도 데이터베이스 통합 작업의 일부분으로 이루어지는 작업이다.

A 회사의 코드시스템과 B회사의 코드 시스템을 통합하려면 A회사의 모든 코드와 B 회사의 모든 코드를 서로 비교하여 가능한 모든 관계를 설정하여야 한다. 모든 관계를 설정하는 작업의 모델을 제시하기 위하여 우선 몇 가지 용어를 통일한다.

코드: 어떤 종류의 데이터의 분류와 조합을 용이하게 수행하기 위하여 사용하는 기호. 예를 들어 회사원 ID 코드, 부품 코드, 행정구역 코드, 등을 들 수 있다.

코드 예: '코드 예'는 주어진 코드의 한 실례를 지칭한다. 예를 들어, 20001230011은 회사원 ID 코드 예가 될 수 있고, LGAC1100은 부품 코드 예가 될 수 있다.

부분코드: 코드를 구성하는 부분을 의미한다. 예를 들어, 사원코드 예, 20001230011에서 2000은 입사연도, 123은 부서, 0011은 일련번호라면 입사연도를 나타내는 연도 코드는 사원코드의 부분코드이다. 마찬가지로 부서 코드도 사원코드의 부분코드다.

코드 영역: 주어진 코드가 취할 수 있는 기호의 범위를 그 코드의 영역이라 한다.

코드 체계: 코드 체계는 어떤 단위 시스템에 사용되는 모든 코드를 일컫는다. 예를 들어, 공정관리 시스템에 사용되는 코드들은 부품코드, 작업장 코드, 사원 코드, 공급자 코드, 납기일자 코드, 등이 포함되며, 이러한 코드를 모두 합하여 공정관리 시스템 코드 체계라고 한다.

본 논문은 코드체계 통합 작업의 모델을 제시한다. 코드체계 A와 코드체계 B를 통합하려면 코드체계 A의 모든 코드1에 대하여 코드체계 B의 모든 코드2 사이에 존재하는 모든 관계를 파악해야 한다.

코드1과 코드2 사이에 존재할 수 있는 관계에는 어떤 것이 가능할까? 모든 관계도 국어로 표현할 수 있다면 모든 관계는 국어 사전의 낱말로 나타난다. 낱말에는 용언과 체언이 있는데 체언은 스스로 주어가 되는 낱말임으로 관계를 나타내는 낱말은 용언이다. 따라서, 국어 사전에 나타나는 용언들을 모두 조사하여 나열할 필요가 있다. 이 과정에서, 같은 의미를 나타내는 용언들은 하나만 나열한다. 즉, 구성지다, 구슬프다, 슬프다, 등은 '슬프다'로, 길다, 기다랗다, 기름하다, 기르스름하다, 길쭉하다, 길쯔막하다, 등은 '길다'로, 덥다, 따뜻하다, 따사롭다, 따스하다, 따습다, 등은 '덥다'만 나열한다. 그러면 다음과 같은 용언들을 나열할 수 있다:

가깝다, 가볍다, 갖다, 같다, 거뿐하다, 거세다, 거치적거리다, 거칠다, 게으르다, 견디다, 겹치다, 경이롭다, 경쾌하다, 고만고만하다, 고지식하다, 곧다, 곱다, 괘씸하다, 괴로워하다, 괴이쩍다, 구성지다, 굳다, 굵다, 굶다, 굼뜨다, 굽다, 급하다, 기껍다, 기쁘다, 길다, 깊다, 까맣다, 까불다, 까칠하다, 깎다, 납작하다, 낮다, 널다, 넓다, 넘다, 넘보다, 높다, 놓다, 뇌다, 늙다, 덥다, 친하다, ...

이러한 용언의 용례 중 A와 B 사이의 관계를 나타내는 것을 살펴보면 다음과 같다:

A는 B를 ... 다.
A와 B는 ...다.
A는 B보다 더 ...다.
A는 B보다 더 잘 ...다.
A가 ... 이면 B는 ...다.

이들 용례의 대표적인 예를 보이면 다음과 같다:

A는 B를 가진다.
A와 B는 친하다.
A와 B는 같다.
A는 B보다 작다.
A가 작으면 B는 크다.

그런데 '코드 예'들은 서로 크기를 비교하는 게 가능하기 때문에 모든 관계를 위의 네 가지로 표현할 수 있다. 즉, 작다, 가볍다, 적다, 굵다, 굼뜨다, 급하다, 까맣다, 등 모든 것이 'A는 B보다 작다'로 표현된다. 왜냐하면 '가볍다'는 무게 관계인데 관계 A와 B가

각각 무게를 나타내는 코드라면 작으면 가볍다고 할 수 있고, '굵다'는 길이를 나타내는 코드를 비교하여 '크다' 혹은 '작다'로 표현할 수 있으며, '굼뜨다'는 속도를 나타내는 코드간의 관계임으로, 이 또한 '작다' 혹은 '크다'로 표현할 수 있다. 따라서, 코드간의 모든 관계는 위에 나열한 대표적인 용례로 다 나타낼 수 있다.

코드1과 코드2간에 성립할 수 있는 관계는 위의 4가지 용례 중 첫째, '코드1이 코드2를 가진다'가 가능하다. '가진다'는 다시 두 가지로 해석이 가능하다. 하나는 '코드2가 코드1의 부분코드'라는 뜻이고, 다른 하나는 '코드2의 영역이 코드1의 영역에 포함된다'라는 뜻이다. 따라서, 이 둘을 명백히 표현하여야 한다. 둘째, '코드1과 코드2가 같다'라는 관계이다. 셋째, '코드1이 ...이면 코드2는 ...이다'라는 관계가 가능하다. 그러나 용례에 보이는 '코드1이 코드2보다 작다'라는 관계는 성립하지 않는다.
따라서 회사A와 B의 코드 통합 작업을 도와주는 시스템은 회사A의 모든 코드와 회사B의 모든 코드의 순서쌍을 차례로 보여주면서, 각 순서쌍에 대하여 위에 보인 가능한 관계, 즉
'코드2가 코드1의 부분코드',
'코드2의 영역이 코드1의 영역에 포함된다',
'코드1과 코드2가 같다',
'코드1이 ...이면 코드2는 ...이다'를 보여주고 시스템 분석자가 실존하는 관계를 기입하도록 한다.

다음은 '코드 예' 사이의 관계에 대하여 살펴본다. "'코드 예 1'은 '코드 예 2'를 가진다", "'코드 예 1'은 '코드 예 2'와 같다", "'코드 예 1'은 '코드 예 2'보다 크다", 등의 일반적 관계는 이미 코드간의 관계에서 다루어 졌고, 개별적인 관계는 별로 의미가 없다.
"'코드 예 1'이 ...면 '코드 예 2'는 ...다"라는 관계는 중요한 역할을 한다. 이러한 관계의 예로, '주민등록번호 코드의 앞자리 8 문자가 19900101이면 생년월일 코드는 1990-01-01이다'와 같은 관계를 들 수 있다. 이러한 '코드 예'간의 관계는 수 없이 많다. 그리고 이러한 관계는 데이터베이스의 자료들이 만족시켜야 할 조건이 된다. 예를 들면, 사원 코드에 성별을 나타내는 부분코드가 있고, 사원 인적 사항 데이터베이스에 성별을 나타내는 속성이 있는 경우, 이들은 서로 반드시 일치해야 한다. 부품 공급자 주소 코드가 경기도라면 전화번호 코드의 지역번호는 031이라야 한다.

코드체계 A와 코드체계 B를 통합하는 과정에서, A의 모든 코드와 B의 모든 코드의 순서쌍과 B의 모든 코드와 A의 모든 코드의 순서쌍에 대하여, 시스템 분석자가 명시한 관계를 행렬에 표시한다. 예를 들어, 코드체계 A가 A1, B1, C1, D1, 등 4 개의 코드로 구성되었다고 하고, 코드체계 B가 A2, B2, C2, D2, E2, 등 다섯 코드로 구성되었다고 하자. '코드2가 코드1의 부분코드'라는 관계를 행렬에 표시하여 <표 1>과 같은 행렬을 구할 수 있다. 이 행렬은 ' 코드 A1은 코드 B2의 부분코드이고', 또한 '코드 B1은 코드 D2의 부분코드이다'라는 관계를 나타낸다.
마찬가지 방법으로 코드 체계 C가 A3, B3, C3라는 코드로 구성되었다면 코드체계B와 C 사이에 성립하는
관계는 <표 2>와 같을 수 있다

	A2	B2	C2	D2	E2
A1		1			
B1				1	
C1					
D1					

.<표 1> 코드체계 A와 B 사이의 부분코드 관계

	A3	B3	C3
A2			
B2			1
C2			
D2			
E2			

<표 2> 코드체계 B와 C 사이의 부분코드 관계

부분코드 관계는 전이관계임으로 코드체계 A와 C 사이의 부분코드 관계는 <표 3>과 같음을 <표 1>의 행렬과 <표 2>의 행렬의 곱으로 쉽게 구할 수 있다.

	A3	B3	C3
A1			1
B1			
C1			
D1			

<표 3> 코드체계 A와 C 사이의 부분코드 관계

<표 1>, <표 2>와 <표 3>에서 A1은 B2의 부분코드이고, B2는 C3의 부분코드이며, A1은 C3의 부분코드임을 알았다.
<표 1>은 코드체계 A에서 B로의 부분코드 관계를 나타낸다. 코드체계 B에서 A로의 부분코드 관계도

역시 행렬로 표시하여야 한다. B2에서 A1로 부분코드 관계가 성립할 수도 있고, C3에서 B2 혹은 A1로 부분코드관계가 성립할 수도 있다. 이러한 부분코드 관계는 A가 B의 부분코드이고 B가 A의 부분코드, 즉 서로 상대방의 부분코드인 관계가 된다. 이러한 관계는 C에서 B로의 부분코드 관계 행렬의 (C3, B2)항이나, B에서 A로의 부분코드 관계 행렬의 (B2, A1)항이 1인가 보면 쉽게 알 수 있다. 부분코드관계는 반사관계다. 즉, 코드 A는 자기 자신, 즉 코드 A의 부분코드다. 그리고 앞에서 본 바와 같이 부분코드관계는 전이관계다. 여기에 서로 상대방의 부분코드가 되면 대칭관계가 성립됨으로 부분코드관계가 동치관계가 된다. 동치관계가 성립되는 코드들의 동치류를 구한다.

마찬가지 방법으로 '코드2의 영역이 코드1의 영역에 포함된다'는 관계도 역시 행렬로 표현하고, 전이 closure를 구한다. 영역의 포함 관계도 역시 반사관계다. 영역의 포함관계는 부분코드관계와 마찬가지로 일반적으로 대칭관계가 아니다. 그럼으로 부분코드관계의 경우와 마찬가지로 코드체계 A에서 B로의 관계에서 B에서 A로의 관계를 알 수는 없다. 즉, B에서 A로의 관계도 역시 명시해야 한다. 영역의 포함관계의 closure를 구하면 부분코드의 경우와 마찬가지로 대칭관계가 성립하는지 살펴본다.

코드 영역의 포함관계에 이어, '코드1과 코드2가 같다'라는 관계를 시스템 분석가로부터 얻는다. '같다'라는 관계는 동치관계임으로 코드체계 A에서 B로 가는 관계만 나타내면 B에서 A로의 관계는 나타내지 않아도 된다. '코드1과 코드2가 같다'라는 관계는 다음과 같은 성질이 있다.

<성질 1> '코드1과 코드2가 같다'의 필요충분 조건은 '행렬의 곱의 결과 코드1과 코드2가 동치류이다'이다.
<증명> 충분조건: 코드1과 코드2가 같으면 '코드1과 코드2가 같다'라고 명백히 쓰여 있든지 아니면 '코드1과 코드i가 같고, 코드i와 코드i+1이 같고, ..., 코드i+k와 코드2가 같다'이다. 행렬의 곱은 전이 closure를 구함으로 두 경우 모두 '코드1과 코드2가 동치류이다'라고 찾는다.
필요조건: '행렬의 곱의 결과 코드1과 코드2가 동치류이다'라면, '코드1' 행, '코드i'열이 1이고, '코드i'행 '코드i+1'열이 1이고, ..., '코드i+k'행 '코드2열'이 1인 일련의 행렬이 존재한다. 이는 코드1부터, 코드i, ... 코드i+k, 코드2까지 일련의 같다는 관계가 성립함을 의미하며, 같다는 관계는 동치관계임으로 '코드1과 코드2가 같다'라고 할 수 있다.

코드간의 관계를 행렬로 표시한 다음 관계가 전이 관계라면 행렬의 곱으로 코드들이 동일한 코드인지 알아 낼 수 있다. 동일한 코드라는 것을 판별하면, 이 사실을 명백하게 알려줌으로써 시스템 관리자가 데이터의 무결성을 검증하는데 사용할 수 있도록 하여야 한다.
관계 중에는 '친하다'와 같이 전이관계가 성립하지 않는 관계도 많이 있다. 즉, A와 B가 친하고, B와 C가 친하더라도 A와 C는 친하지 않은 경우가 있다. 전이 관계가 성립하지 않는 경우에는 행렬의 곱 연산을 수행하는 것은 아무 의미가 없다.
코드체계 통합에서 가장 중요한 과정은 각 코드간에 '코드1이 ...이면 코드2는 ...이다'라는 관계와 "'코드 예 1'이 ...면 '코드 예 2'는 ...다"라는 관계를 파악하는 것이다. 앞에서 주민등록번호와 생년월일 관계에서 언급한 바와 같이 두 코드간에 밀접한 관계가 있으면 데이터베이스에 갱신이나 삽입 작업이 수행될 때마다 이러한 관계가 만족되는지 자동적으로 검사해야 한다. 따라서, 이러한 관계를 코드체계 통합 과정에서 명백히 발견하여 시스템 관리자에게 문서로 남겨 줌으로써 데이터베이스 관리에 적용할 수 있도록 해야 한다.

4. 결론

여러 시스템을 통합해야 하는 경우가 많이 있다. 회사가 합병하는 경우에는 물론이려니와 부서마다 독자적으로 개발한 시스템들을 통합하여 표준화함으로써 CALS와 EDI에 대비해야 한다. 시스템 통합 과정에 빼놓을 수 없는 것이 데이터베이스 통합이며, 데이터베이스 통합의 일환으로 코드체계를 통합하여야 한다.
본 논문은 코드체계 통합과정의 모델을 제시하였다. 즉, 코드체계 통합과정에서 각 코드간의 다음과 같은 관계를 파악하여야 한다:
'코드2가 코드1의 부분코드',
'코드2의 영역이 코드1의 영역에 포함된다',
'코드1과 코드2가 같다',
'코드1이 ...이면 코드2는 ...이다',
"'코드 예 1'이 ...면 '코드 예 2'는 ...다"
코드체계 통합 도구는 시스템 분석자가 이러한 관계를 명시하기 쉬운 환경을 제공하고, 명시한 관계를 분석한 다음, 문서화한다. 코드간의 관계를 나타내는 이 문서는 데이터베이스 관리자가 데이터베이스의 일관성을 유지하기 위한 작업을 수행하는 데 중요하게 사용된다.

5. 참고 문헌

[1] 최창원, 윤성현, "EDI(전자식 자료 교환) 기술의 특성", 정보처리학회지, vol.4, No.1, 1997년 1월, pp.44-57.