

데이터 코드 생성 지원 전문가 시스템의 설계

박대하, 정인기, 백두권
고려대학교 전산학과

요 약

정보화 사회에서 대량으로 생산된 데이터 코드들은 일관된 설계 원칙없이 필요할 때마다 만들어 사용함으로써 정보의 중복 저장 및 정보 교환에 있어서의 변환 작업 등으로 인한 경비의 소요가 상당한 실정이다.

이러한 문제점에 대한 해결책으로 본 논문에서는 데이터 코드 설계자가 일관성있게 데이터 코드를 생성할 수 있도록 도와주는 데이터 코드 생성 지원 전문가 시스템의 설계에 관하여 연구하였다.

불완전 영역 설계를 위한 지식 획득과 표현에 적합한 전문가 시스템 셸인 GUESS(Guideline Underlying Expert System Shell)를 설계하였다. GUESS는 전문가 시스템을 설계 지원 도구로 사용하는 사용자에게 기존에 작성된 적절한 설계 용례를 선택의 기준으로 제공하며, 유연성있는 작업 지침들을 규칙으로 포함하고 있다. GUESS는 Prolog 언어를 기반으로 한 추론 기관과 설계 지침을 포함하는 정적 지식, 외부 데이터베이스를 연결한 동적 정보, 설계 세부 방법을 담고 있는 부가 도구들로 구성된다.

GUESS/DCG는 데이터 코드 생성을 지원하기 위하여 데이터 코드의 유형과 선택 기준 및 설계 원리를 정적 지식으로 가지며, 이를 경험적으로 탐색하는 추론 기관 및 사용자인 데이터 코드 설계자와 적절한 대화식 접근을 가능하게 하는 설명 부분과 대화 인터페이스를 GUESS를 바탕으로 구현한 것이다. 특히 동적 정보의 적절한 이용과 데이터 코드의 통합된 저장, 일관성있는 운영을 보장하기 위하여 개발중인 데이터 코드 관리 시스템과의 인터페이스 부분을 추가하여 기존에 운영되고 있는 데이터 코드의 참고와 호환성, 확장성을 유지하였다.

이 시스템은 데이터 코드 관리 시스템에 일관된 생성 수단을 제공하는 것 외에도, 각 기관에서 대량으로 작성되는 데이터 코드를 유지, 보수하는 작업에도 큰 기여를 할 것이다.

I. 서론

현대 사회의 급속한 정보화 흐름에 따라 대량의 정보가 창출되었으며, 이를 사용하는 조직간에 정보 교환의 필요성이 대두되었다. 특히 회사, 행정 부처 등과 같은 조직의 정보 자원 관리와 처리를 위한 공통 데이터 코드의 개발은 일관되고 통일된 자원 이용의 효율을 높이기 위한 공통 코드의 개발이 요청되고 있는 실정이다[1].

또한 일관된 설계 원칙없이 작성된 데이터 코드는 정보의 교환과 중복 저장등의 문제점을 가져오므로 기존의 데이터 코드와 호환성 및 일관성을 유지하며 잘 정의된 설계 방법에 의한 데이터 코드 생성을 책임지는 데이터 코드 생성 지원 도구의 구축이 필요하다.

데이터 코드는 유일성, 확장가능성, 간결성, 단순성 등의 일반 측면을 가지며, 데이터 요소로 분류될 수 있는 모든 정보는 코드화될 수 있다[2]. 코딩되는 자료가 정보 구조 안에서 갖는 역할에 따라서 표현 형태가 각기 달라질 수 있으며, 데이터 코드가 표현하는 정보의 종류가 무한하다는 점에서 생성 작업의 어려움이 발생한다. 이 어려움을 해결하고 실제 데이터 코드를 설계하는 설계자에게 데이터 코드 생성을 도와주기 위한 도구로서 전문가 시스템의 개발하게 되었다.

정해진 지식에 의하여 특수 영역에 적합한 결과를 출력하는 설계 지원 전문가 시스템과 달리 데이터 코드 생성을 지원하는 전문가 시스템은 모든 정보에 걸친 불완전한 영역의 설계를 담당하므로 특별한 형태의 지식 표현이 필요하다.

본 연구에서는 제한된 지식을 기반으로 넓은 범위의 영역에 대한 설계를 가능하도록 불완전 영역 설계를 위한 전문가 시스템 쉘인 GUESS를 제안하였다. 또한 데이터 코드 설계 방법론을 지식 베이스로 하여 데이터 코드 관리 시스템과 인터페이스를 갖는 데이터 코드 생성 지원 도구인 GUESS/DCG를 개발하였다.

본 논문에서는 먼저 데이터 코드 설계 방법론과 데이터 코드 생성 지원 도구의 필요성을 서술하고, GUESS의 구조적 특징을 살펴본 후, 이를 통합한 GUESS/DCG의 구성을 제시하였다.

II. 데이터 코드 설계 방법론과 생성 지원 도구

1. 데이터 코드의 유형과 선택 기준

데이터 코드를 설계 유형에 따라 분류하면 그림 1과 같다[1][2]. 무의미 코드, 의미 코드, 논리 코드, 대조 코드, 약어 기반 코드와 분류 코드는 데이터 코드 설계 방법을 분류하기 위한 의미의 용어들이고, 그 외에 이중 테두리를 갖는 코드들이 실제 데이터 코드로 생성될 설계 기법에 대한 유형이다.

어떤 유형의 코드를 선택하는가는 사용되는 용도에 따라서 데이터 코드 설계자의 주관에 맡겨진다. 일반적으로 조합 코드 또는 기호 코드가 가장 널리 사용되며 순차 코드나 임의 코드와 같은 무의미 코드는 분류 코드를 포함하는 복합 코드의 일부로 사용된다. 기록상의 오류 검사 능력을 제공하는 자기 검증 코드는 모든 유형의 데이터 코드에 추가적으로 사용될 수 있다.

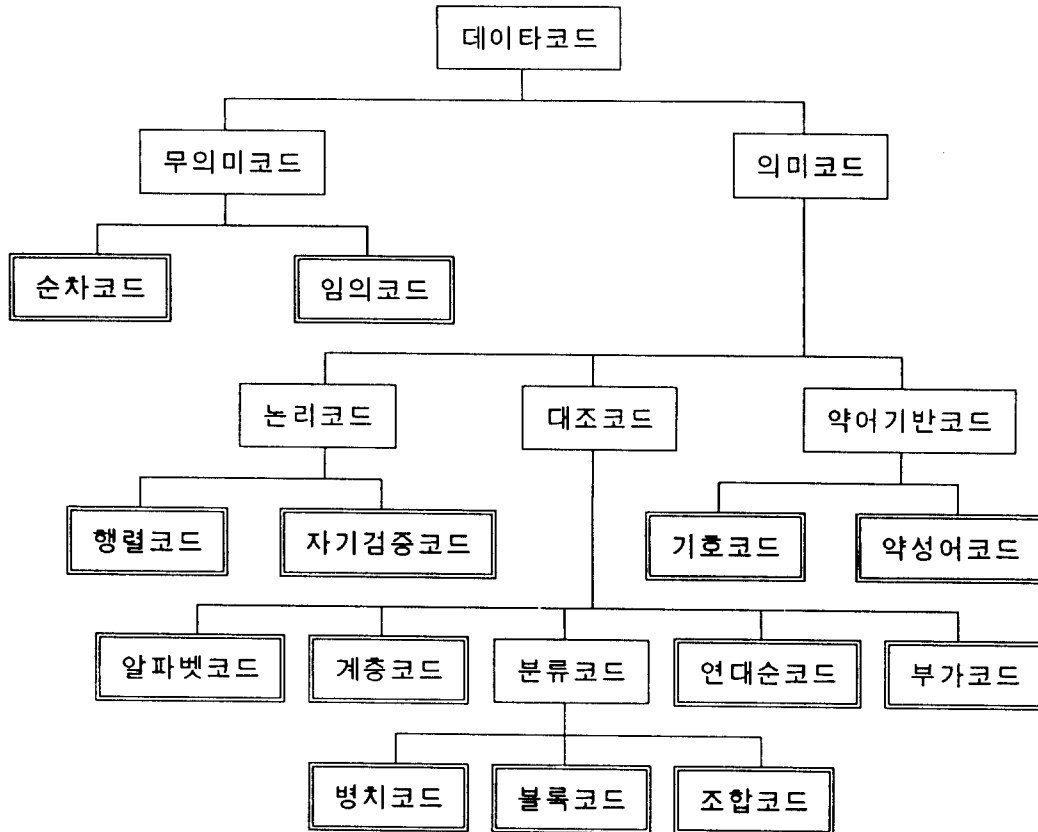


그림 1 데이터 코드의 유형별 분류

데이터 코드 설계자가 적당한 데이터 코드의 유형을 선택하기 위한 기준으로서 <선택예 1> ~ <선택예 7>을 들 수 있다.

- <선택예 1> 코드 자체에 의미를 부여할 것인가?
-> 의미 코드 / 무의미 코드 선택
- <선택예 2> 데이터 요소가 분류될 수 있는가?
-> 분류 코드 선택
- <선택예 3> 데이터 요소의 약어가 이미 존재하는가?
-> 약어 기반 코드 선택
- <선택예 4> 데이터 요소가 계층을 이루고 있는가?
-> 계층 코드 또는 조합 코드 선택
- <선택예 5> 데이터 요소간에 순서가 있는가?
-> 순차 코드 선택
- <선택예 6> 데이터 코드가 여러 요소의 조합으로 이루어질 수 있는가?
-> 행렬 코드 또는 분류 코드 선택
- <선택예 7> 같은 의미로 사용되는 표준 이외의 코드가 존재하는가?
-> 다중 코드 선택

선택예를 뒷받침해 줄 수 있는 용례가 기존에 존재한다면 더욱 선택이 편리할 것이다. 예 1.과 예 2.는 기호 코드와 계층 코드의 실제 용례이다.

예 1. 기호 코드 (국가명을 표현하는 ISO 3165 코드에서)

<u>개체명</u>	<u>국가 코드</u>
Austria	AT
Canada	CA
France	FR
Switzerland	CH
United Kingdom	GB
United States	US

예 2. 계층 코드 (Harmonized system 코드 형식에서)

37	사진 그리고 영화 상품
3705	영화필름이 아닌 사진 인화지와 인화, 현상된 필름
370520	마이크로필름

2. 데이터 코드 설계 원칙

데이터 코드의 설계 유형에 대한 선택만으로 완전한 데이터 코드 집합을 생성한다는 것은 상당히 힘든 작업이다. 그러나 각 데이터 코드의 유형에 따라 고유한 설계 방법이 존재함은 물론이고, 데이터 코드 설계 전반에 걸친 기본 원칙이 국제 규격으로 제정되어 있다[2][3][4][5]. 일반적으로 적용되는 몇 가지 중요한 원칙을 살펴보면 다음과 같다.

2.1 기존 코드의 이용

완전히 새로운 체계의 코드는 절대적으로 필요한 상황이 아닌 경우에는 설계하지 않도록 한다. 많이 사용되는 기본적인 데이터 요소에 대한 코드는 국제 코드(ISO 규격, ECE 추천안 등)를 따라야 한다.

국제 표준, 국내 표준, 기업 표준, 조직 표준, 그리고 응용 표준 코드 등의 순서로 계층적 구조를 이루어 상위 표준과 중복을 줄이고 일관성을 유지한다.

2.2 코드 의미성의 부여

의미 코드가 적절히 사용되게 되면 정보의 첨가에 도움을 줄 수 있으며, 무의미 코드보다 사용자에게 용이성을 제공할 뿐만 아니라 신뢰성을 부여한다. 그러나 과중한 의미 코드의 부여는 자유로운 조작을 힘들게 할 뿐만 아니라 확장성의 결여를 가져오므로 피해야 한다.

2.3 문자 수의 고정

데이터 코드의 값은 데이터 통신 시간을 줄이고 공백을 줄이기 위해 문자의 수를 최소화할 수 있도록 구성되어야 한다. 그러나, 동시에 코드 사용자의 능력에 맞게 최적화되어야 한다. 특히 응용에 사용될 시스템의 처리 한계를 넘어서는 크기의 숫자나 문자는 피한다.

고정된 문자수를 가진 코드는 가변 길이 코드보다 신뢰성이 있고 사용하기 편하다. 또한 4 개의 알파벳 또는 5 개 이상의 숫자와 문자를 합성한 코드값은 그보다 적은 수의 코드 부분으로 나누어 신뢰성있는 기록을 가능하도록 분리되어야 한다. (예, ABC-12-3456은 ABC123456보다 신뢰성이 있다.)

2.4 코드 용량의 산출

코드 용량은 선택된 숫자를 기저로 하여 각 위치에 대한 모든 이용가능한 문자의 조합으로 생겨나는 코드 표현의 수로 나타낼 수 있다. 이 용량은 이론적인 것으로서 모든 문자의 가능한 조합을 사용한다는 것을 전제로 한 것이다.

코드의 독자성을 유지하면서 모든 상황을 표현하도록 주어진 코드의 용량을 계산할 때, 다음 공식이 적용된다. (영문자 I와 O는 가능한한 피해야 하므로 24 개의 알파벳과 10 개의 숫자를 사용한 경우이다.)

$$C = (24^A)(10^N)$$

여기서

C = 이용가능한 총 코드의 조합

A = 코드 내의 알파벳 문자 자릿수

N = 코드 내의 숫자 자릿수

(A + N은 코드의 전체 자릿수와 같다.)

3. 데이터 코드 생성 지원 도구

3.1 데이터 코드 생성 지원 도구의 필요성

대부분의 데이터 코드는 기업과 행정 부처에서 일정한 설계 원칙없이 필요에 따라서 생성되며, 코드 설계자의 주관에 의존하여 같은 데이터 요소를 코드화하여도 다른 유형의 코드가 만들어지는 경우가 허다하다. 또한 국제 규격으로 제정된 데이터 코드 설계 방법론에 의한 코드 생성도 설계자의 경험이 미숙하면 많은 시행착오를 겪게 된다.

여기에서 일관된 데이터 코드 설계 방법론을 내장하여 데이터 코드 설계자의 작업을 도와주는 자동화된 데이터 생성 지원 도구의 필요성이 발생한다.

본 논문에서는 국제 규격에 따른 데이터 코드 설계 방법론을 분석하여 이를 지식 베이스화한 데이터 코드 생성 지원 전문가 시스템을 설계하여 이러한 필요성을 만족시키는 방안으로 제시하고자 한다.

3.2 설계 방법론의 지식 표현시 고려사항

이상에서 살펴본 것과 같은 데이터 코드 설계 방법론이 전문가 시스템에 사용되는 지식 베이스로 표현되기 위해서는 다음과 같은 사항이 고려되어야 한다.

- 1) 데이터 코드 생성을 위한 일련의 처리 과정이 순서있게 선언되도록 한다. 코드 유형의 선택, 기존 코드의 참조, 코드 세부 설계, 설계된 코드의 등록 과정을 순차적으로 진행한다. 각 과정은 상하 단계로 구분되어 필요에 따라서 상위 단계로 거슬러 올라갈 수 있다.
- 2) 코드 유형 선택을 위한 규칙과 설계 원칙을 포함하는 정적 지식의 구축이 필요하다. 규칙 번호를 할당하여 설명 부분과 코드 생성 부가 도구를 호출할 때 이용한다.

- 3) 사용자인 데이터 코드 설계자가 설계 과정을 이해하기 쉽도록 대화식 사용자 인터페이스를 가진 설명 기능이 제공되어야 한다.
- 4) 데이터 코드의 세부 설계시 공통적으로 사용되는 순차 코드 할당, 코드 용량 산출 등을 지원하기 위하여 코드 생성 부가 도구의 개발이 필요하다. 데이터 코드 관리 시스템 인터페이스에 제공될 중간 코드를 생성하는 루틴도 부가 도구에 포함시킨다.
- 5) 데이터 코드 관리 시스템에서 제공되는 기존 코드에 관한 정보를 저장하거나 데이터 코드 생성 과정의 현재 위치를 기록, 갱신하기 위하여 동적 정보가 유지되어야 한다.

Ⅲ. GUESS : 불완전 영역 설계를 위한 전문가 시스템 셸

1. GUESS의 구성

GUESS(Guideline Underlying Expert System Shell)는 지침을 포함하는 순차적인 설계 단계와 각 단계별 세부 규칙을 갖는 정적 지식, 현재 진행 중인 정적 지식의 추론 단계와 외부 데이터베이스의 내용을 임시로 저장하여 설계 용례를 제공하는 동적 정보, 그리고 실제 설계 작업과 외부 데이터베이스 인터페이스를 담당하는 부가 도구가 추론 기관을 중심으로 연결된 형태로 구성된다(그림 2).

2. 추론 기관의 설계

GUESS는 후진 추론(backward chaining) 기관을 내장하고 있는 Prolog 언어를 기반으로 설계되었으나, 기본적인 추론 형태는 생성 시스템(production system)을 사용한 전진 추론(forward chaining)이다.

후진 추론에 의한 목표 지향 접근방식(goal driven approach)는 의학 진단 시스템과 같이 도달 가능한 해답이 많지 않을 경우에 효율이 있다[5]. 그러나 본 시스템과 같이 불완전한 영역 내에서 가능성있는 무수한 해답 중에서 하나를 얻기 위해서는 전진 추론을 사용한 데이터 지향 접근방식(data driven approach)이 적합하다.

GUESS에 사용된 전진 추론 규칙은 설계 단계(Level)와 세부 규칙(Rule)을 이용하여 다음 설계 단계로 진행하거나 필요로 하는 부가 도구(Tool)에 접근하는 방식이다.

```

<Level-Rule (n, i)>
=> [ GO      : <Level-Rule (m, j)> ]
    [ USE     : <Tool k> ]
    [ EXPLAIN : <Info (n, i)> ]           규칙 (1)

```

규칙 (1)에서, 상부 단계에서 하부 단계로 진행하기 위하여 $m = n - 1$ 을 사용하며, 설계 과정을 재출발하기 위해서 m 을 0으로 리셋시키면 된다.

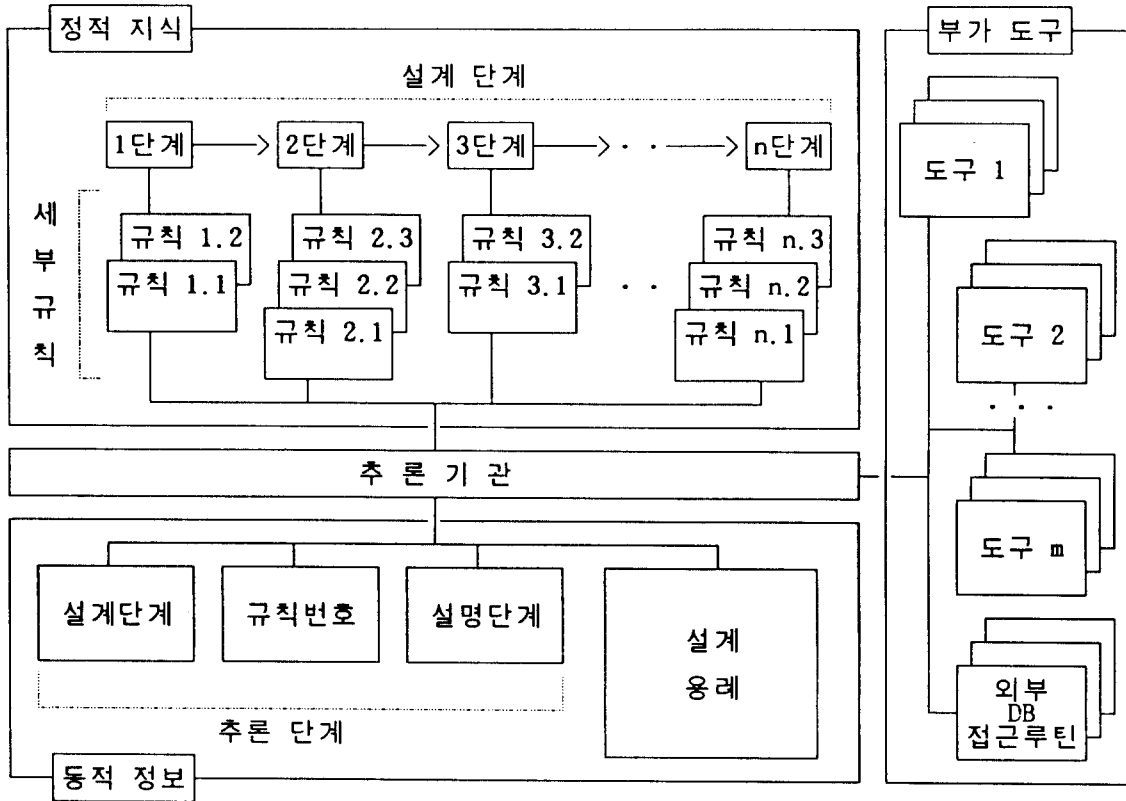


그림 2 GUESS의 구조

3. 정적 지식과 동적 정보의 구성

정적 지식(static knowledge)은 일련의 설계 단계로 이루어지며, 각 설계 단계마다 다수개의 세부 규칙을 포함하고 있다.

설계 단계에서 적용되는 지침을 만족하면 세부 규칙 중 하나를 선택하게 선택하게 되고, 선택된 세부 규칙을 수행한 후 하부 설계 단계로 이동하게 된다.

규칙 (2)와 (3)은 각각 설계 단계와 세부 규칙을 나타낸다.

```
<Guideline (n,1)> and <Guideline (n,2)> ... and <Guideline (n,m)>
=> [ GO : <Rule i> ]                                규칙 (2)
```

```
<Rule i>
=> [ USE : <Tool k> ]
    [ EXPLAIN ]                                        규칙 (3)
```

동적 정보(dynamic information)는 설계 단계와 규칙 번호, 그리고 설명 단계를 포함하여 추론 기관이 현재 진행 중인 과정을 유지한다. 외부 접근 루틴에 의하여 생성된 설계 용례를 포함하여 일반적인 전문가 시스템의 작업 공간의 확대된 형태라 할 수 있다.

IV. GUESS/DCG : 데이터 코드 생성 지원 전문가 시스템

1. 데이터 코드 설계 원칙의 추가

데이터 코드 생성을 위한 일련의 순서 단계를 GUESS의 설계 단계에 추가하여 데이터 코드 생성 지원 전문가시스템인 GUESS/DCG(GUESS for Data Code Generation)를 구성하였다(그림 3). 본 논문 II장의 3.1에서 고려사항으로 제시된 단계에 동적 정보를 재설정하는 초기화 단계를 추가하여 그림 4의 설계 단계로 순서지을 수 있다.

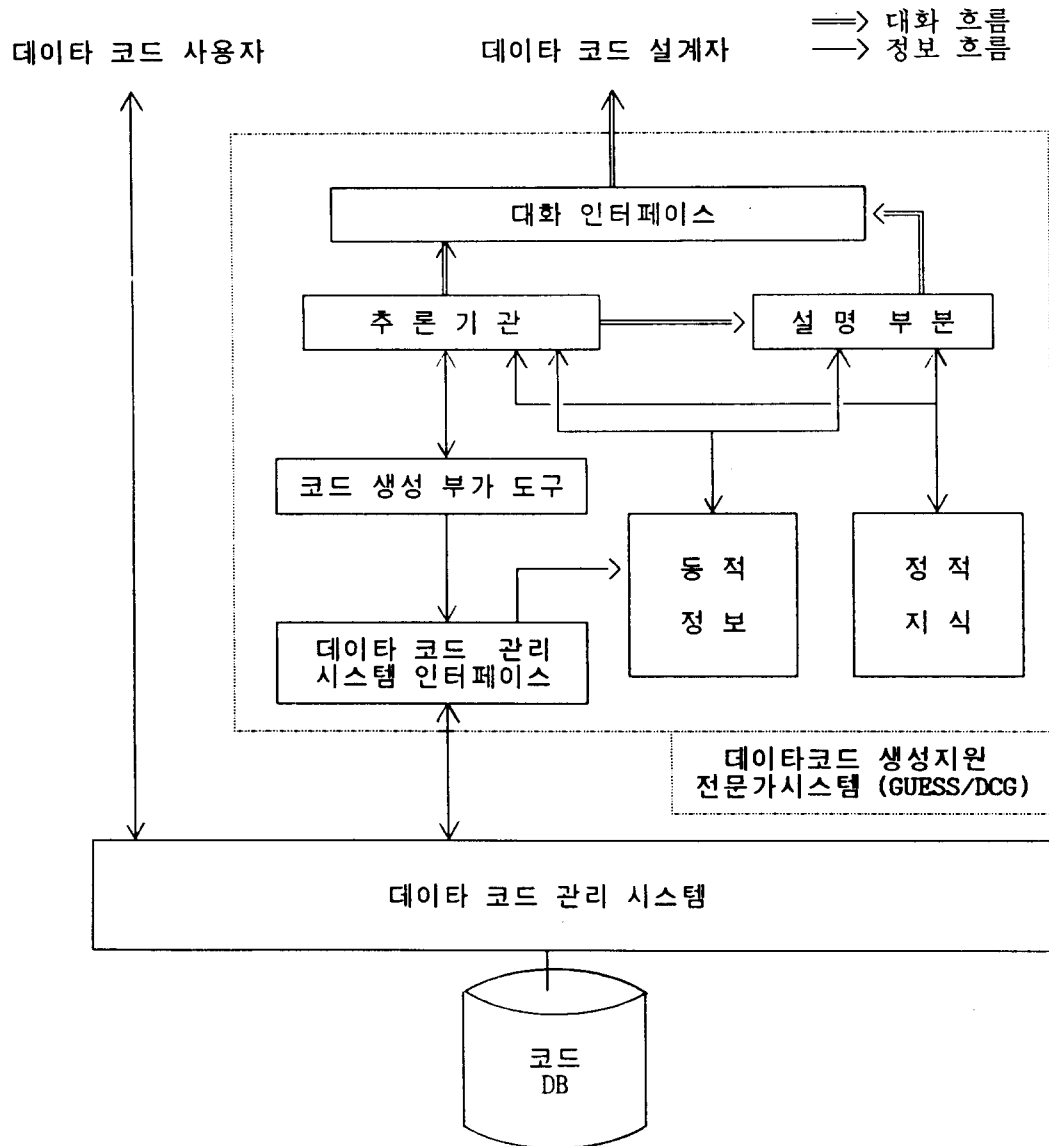


그림 3 데이터 코드 생성 지원 전문가 시스템의 구성

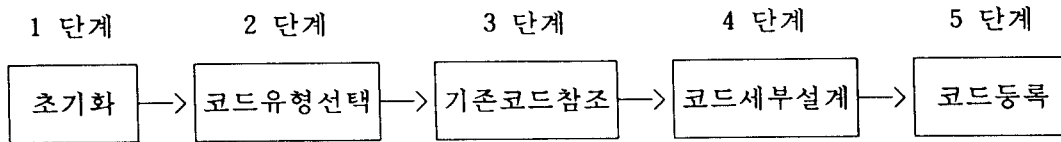


그림 4 데이터 코드 설계 단계

2. 데이터 코드 관리 시스템 인터페이스

코드 생성 부가 도구를 거쳐 생성된 데이터 코드를 데이터 코드 관리 시스템에 저장하고, 데이터 코드 관리 시스템에서 추론 기관이 원하는 항목을 조회할 수 있도록 외부 함수를 호출하는 인터페이스를 갖도록 한다.

3. 설명 부분과 대화 인터페이스

GUESS/DCG는 사용자인 데이터 코드 설계자에게 설계 단계별로 유연한 접근 형태를 제공하기 위하여 II장의 1에 나타난 예 1, 2와 같은 대량의 설계 용례를 참조할 수 있다. 설계 용례는 "EXAMPLE"이란 사용자 질의에 따라서 데이터 코드 관리 시스템 인터페이스에 의하여 동적 지식에 저장된다. 설명 부분은 화면 출력에 적당한 형태로 설계 용례를 변형한 후 사용자에게 제공한다.

설명 부분을 거쳐 생성된 정적 지식에 따른 작업 과정과 동적 지식에 담긴 설계 용례 사용자에게 제시하고, "WHY", "WHAT" 등의 사용자 질의를 입력받기 위하여 대화 인터페이스를 구축하였다.

V. 결론

본 논문에서는 데이터 코드 생성을 지원하는 전문가 시스템의 설계에 관한 전반적인 사항을 다루었다.

불완전한 설계 영역을 다루기 위하여 일정한 지침만을 내재하고 대량의 외부의 정보를 이용할 수 있도록 GUESS를 설계하였으며, 데이터 코드 생성을 위한 원리를 정적 지식에 추가하여 전체적인 시스템을 구성하였다. 사용자 위주의 대화 인터페이스와 데이터 코드 관리 시스템에 대한 인터페이스 부분에 잘 정의된 상호 작용이 앞으로 연구되어야 할 것이다.

무분별하게 대량으로 생산된 정보를 통합하기 위하여 EDI와 같은 사후 대책이 발생하는데 비하여, 본 시스템에서 제공된 일관된 설계 방법론에 의하여 설계된 데이터 코드를 유지하는 데이터 코드 관리 시스템은 정보 통합에 대한 조기 치료를 제공할 수 있다.

본 시스템은 SUN Workstation에서 SB-Prolog를 사용하여 구현중이다.

참고문헌

- [1] 한국전산원, 「국가기간 전산망 공통 데이터 표준 개발」, 1991.
- [2] ISO/IEC/JTC1/SC14, *Guidelines for Organization and Representation of Data Elements for Data Interchange - Coding Methods and Principles*, CDTR 9789, N607, 1992.03.18.
- [3] FIPS PUB 45, *Guide for the development, Implementation and Maintenance of standards for the Representation of Computer Processed Data Element*, 1976.
- [4] ISO/IEC/JTC1/SC14, *Basic Attributes of Data Elements (Types)*, CD11179, N599, 1991.12.06.
- [5] ISO/IEC/JTC1/SC14, *Framework for the Generation and Standardization of Data Elements (Types)*, WD, N610, 1992.04.28.
- [6] Dennis Merritt, *Building Expert Systems in Prolog*, Springer-Verlag, 1989, pp.73 - 96.
- [7] J.Ross Quinlan, *Applications of Expert Systems*, Addison Wesley, 1987.
- [8] J.Ross Quinlan, *Applications of Expert Systems Vol 2.*, Addison Wesley, 1989.
- [9] Steven S.Muchnick, Peter Schnupp, *Expert Systems Lab Course*, Springer-Verlag, 1989.
- [10] Prederick Hayes-Roth, Donald A.Waterman, Douglas B.Lenat, *Building Expert Systems*, Addison Wesley, 1983.
- [11] Atsuo Kawaguchi, Hiroshi Motoda, Riichiro Mizoguchi, *Interview-Based Knowledge Acqusition Using Dynamic Analysis*, IEEE EXPERT, October 1991, pp.47 - 60.
- [12] William Swartout, Cécile Paris, Johanna Moore, *Design for Explainable Expert Systems*, IEEE EXPERT, June 1991, pp.58 - 64.
- [13] James R.Slagle, David A.Gardiner, Kyungsook Han, *Knowledge Specifica-tion of an Expert System*, IEEE EXPERT, AUGUST 1990. pp.29 - 38.