KAIS '97 추계학술대회

CAPP 지원을 위한 사례베이스의 구조화

김진백
부산여전 경영정보과

김유일
부산대학교 경영학과

요약

사례기반형 추론(CBR)은 과거의 경험을 이용해서 문제를 해결하려는 방법으로서 규칙기반형 추론(RBR)과 달리 문제해결경험이 풍부한 도메인에 적합한 방법이다. CBR은 정적인 측면에서 사례의 표현과 구조화문제가 중요시 되며, 동적인 측면에서는 사례의 검색절차와 수정이라는 해결안 생성과정이 중요시된다. 본 논문은 정적 측면에서 효과적인 CAPP 지원을 위해 사례베이스(CB)를 계층적으로 구조화하였다. 또한 CB의 구조화시 시스템의 문제해결 능력을 향상시켜주기 위하여 CB를 응용도메인 종속적 CB(DDCB)와 독립적 CB(DICB)로 분리하여 과거의 문제해결 경험에 관한 지식은 DDCB에 나타내었으며, 도메인 전문가가 가지는 일반적인 문제해결 지식은 DICB에 나타내었다.

Ⅰ. 서론

정보기술을 활용한 의사결정 지원도구들에 대한 관심은 초기의 데이터베이스 시스템에서 모형관리시스템으로 이어지면서 다양한 형태의 정보시스템들이 등장하였다. 최근에는 보다 효율적인 의사결정지원을 위해 AI 분야에 대한 관심이 고조되고 있다. AI 시스템은 지식집약적 시스템(knowledge-intensive system)으로 시스템의 지식은 인간 전문가로부터 지식공학자에 의해 획득되거나 혹은 시스템의 학습기능에 의해 시스템 자체가 획득할 수 있다. 규칙기반형 시스템(Rule-Based Systems : RBS)은 전자의 방법에 의해 지식이 획득되는 대표적인 시스템으로서, 시스템이 이용하는 지식은 생성규칙(production rule)의 형태로 저장되는 것이 일반적이다. 후자의 방법은 인간 전문가로부터의 지식 획득이 용이하지 않은 경우 혹은 지식획득의 효율성을 높히기 위해 활용되는 방법으로 이의 대표적인 시스템이 사례기반형 추론시스템(Case-Based Reasoning Systems : CBRS)이다. CBRS는 과거의 의사결정 경험을 이용해서 문제에 대한 해결책을 제시하려는 시스템으로서 RBS보다 구현과 유지보수 측면 등에서 많은 이점을 가지고 있다(Gupta, 1994). 그리고 CBRS는 지식획득과정에 대한 병목현상을 해결해줄 수 있는 이점을 가지고 있어 활용 범위가 넓어질 것으로 예상되고 있다(Mott, 1993).

CBRS에서는 문제해결에 필요한 자료와 모형을 사례형식으로 표현해서 사례베이스(CaseBase : CB)에 저장한다. CBRS가 효과적으로 의사결정을 지원하기 위해서는 각 사례의 표현문제와 사례들의 활용문제가 중요 연구과제가 된다. 본 논문은 정적 측면에서 사례 표현에 대한 요구사항들과 공정계획문제를 대상으로 사례의 효율적 저장과 연결을 위해 CB의 구조화문제를 다룰 것이다.

Ⅱ. 사례기반형 추론과 사례표현

1. 추론방법들의 지식표현 형태 및 적용가능영역

사례기반형 추론(Case-based Reasonig : CBR)은 추론에 이용되는 지식의 형태, 내용 및 적용조건에 있어서 규칙기반형 추론(Rule-Based Reasoning : RBR) 및 모형기반형

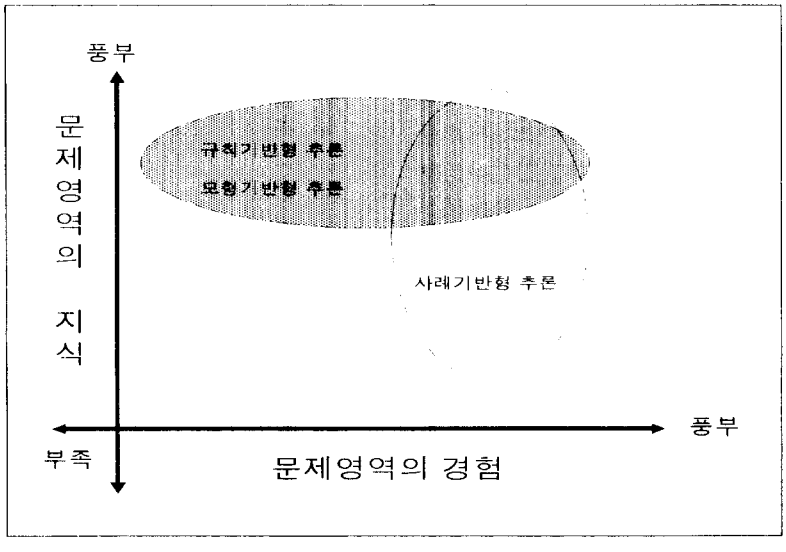
추론(Model-Based Reasoning : MBR)과 차이가 있다. RBR은 AI 분야에서 이용되는 추론기법중 대표적인 것이라 할 수 있으며 "IF … THEN …" 형식의 생성규칙으로 지식을 표현한다. 그러나 CBR은 문제해결에 관련된 경험을 사례형식으로 표현한다. 그리고 CBR은 과거의 반복적인 경험을 통해 문제해결방법을 습득하는 방법으로서 일종의 귀납적 추론이라 할 수 있다. 따라서 CBR의 대상영역은 경험이 풍부한 영역이 되는 것이 일반적이다. RBR은 잘 정의된 지식베이스를 이용해서 문제를 해결하려는 연역적 추론법이므로 지식이 풍부한 영역에 대한 문제해결방법으로서 적합하다. RBR은 지식이 부족한 영역에 대해서 문제해결능력이 제한적이며, CBR은 경험이 부족한 영역에 대해서 문제해결능력이 제한적이라 할 수 있다. RBR은 응용분야에 대한 명확한 지식이 풍부할 경우에 유용한 추론기법이 될 수 있으나 지식의 획득에 대한 과다한 소요시간, 유지보수의 곤란, 설명 및 학습능력의 부족, 지식표현의 한계, 그리고 강건성 결여 즉, 해가 없는 경우와 질의가 잘못된 경우를 구별하지 못하고 모두 해가 없는 것으로 인식한다는 문제점이 있다(Mahapatra 등, 1994; Vargas 등, 1993; Slade, 1991).

MBR은 원인과 결과에 대한 모형을 이용해서 지식을 표현한다. MBR은 인과모형이 존재할 경우에 적용가능하기 때문에 RBR과 같이 응용분야에 대한 충분한 이해가 필요하다. CBR은 응용분야에 대한 충분한 이해가 없더라도 일정한 조건하에서 해당 사례를 적용할 수 있다. 따라서 사례집합은 응용분야에 대한 이해가 불충분한 경우 일반화된 모형의 역할을 수행한다. MBR은 해결안을 검증할 수 있는 수단을 제공해주지만 해결안 생성은 지원하지는 않는다. 그러나 CBR은 효율적인 해결안 생성을 지원하며 해결안의 평가는 최적 사례에 근거를 둔다.

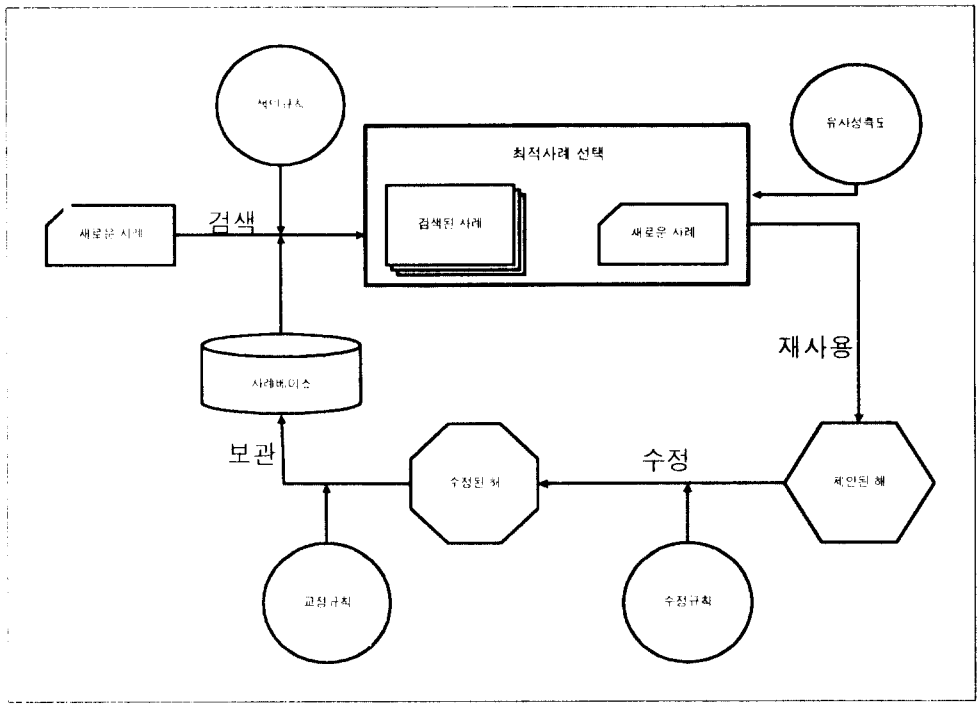
따라서 이들 추론방법들의 적용가능영역은 <그림 1>과 같다. 만약 문제해결영역에 대한 경험과 지식이 모두 풍부한 경우에는 세가지 추론방법 모두가 적용가능하나, 문제해결영역에 대한 경험 혹은 지식이 한가지만 풍부할 경우에는 CBR 혹은 RBR 및 모형기반형 추론방법이 적합할 것이다. CBR이 이들 두가지 추론방법들의 효과적인 대체수단이 될 수 있는 영역은 경험과 지식이 풍부한 문제해결영역이라 할 수 있다.

2. 사례기반형 추론의 과정

CBR은 새로운 문제가 주어지면 이를 분석해서 과거의 경험중 가장 유사한 경험을 검색한 후 이를 직접 혹은 부분 수정해서 문제를 해결하므로서 의사결정의 효율성을 증대시키게 된다. Slade(1991)는 CBR 과정을 문제



<그림 1> CBR과 RBR의 적용가능영역



<그림 2> 사례기반형 추론의 과정모형

의 속성에 대한 색인부여, 검색, 수정, 테스트 및 저장이라는 5 단계로 나눈 후 실패에 대한 설명 단계를 추가해서 총 6 단계로 나누었다. McGovern 등(1994)은 CBR 과정을 문제의 명세화, 검색, 선택, 해결안의 구축, 해결안의 실험, 해결안의 시행, 해결안의 평가 및 사례 기억장치의 갱신이라는 8 단계로 세분화하였다. 본 논문에서는 CBR의 과정을 Aamodt 등(1994)이 나눈 4 단계 모형인 4 REs를 통해 살펴보기로 한다.

① 검색(REtrieve) : 새로운 문제가 주어지면 사례기반형 추론은 문제의 주요 속성을 추출한 다음 이들 속성을 이용해서 과거의 유사한 사례를 검색하게 된다. 검색결과는 3가지로 요약될 수 있으며, 첫째는 둘 이상의 유사한 사례가 검색되는 경우, 둘째는 유일한 유사 사례가 검색되는 경우, 셋째는 유사한 사례가 발견되지 않을 경우이다.

② 재사용(REuse) : 1단계의 검색결과 단일 사례가 검색되면 이를 통해 문제를 해결하게 되며 둘 이상의 유사한 사례가 검색되면 속성별로 비교를 통해 유사성을 측정한 후 주어진 문제와 가장 유사성 측정치가 높은 사례를 기저 사례(base case)로 선택해서 새로운 문제의 해결을 시도한다.

③ 수정(REvise) : 2단계를 통해 선택된 기저 사례가 새로운 문제와 완전히 일치할 수도 있으나 경우에 따라서는 부분적인 차이가 발생할 수도 있다. 따라서 이 경우에는 사용자와의 상호작용을 통해서 차이 부분에 대한 부분적인 수정과정을 거쳐서 새로운 문제에 적합한 해를 생성시키게 된다.

④ 보관(REtain) : 새로운 문제가 기존의 사례와 완전히 일치하였을 경우에는 새로운 문제의 해결방법을 저장한다는 것은 중복저장이 되므로 이 경우는 새로운 문제의 해를 저장할 필요가 없다. 그러나 새로운 문제의 해가 기존의 사례와 다소의 차이가 있을 경우에는 새로운 문제의 해를 사례베이스에 저장해서 향후 의사결정에 이용할 수 있게 한다.

이상의 4 REs 과정은 주기성을 띠는 것으로서 사례기반형 추론의 과정을 지원하기 위

해 색인규칙(indexing rules), 사례 기억장치(case memory), 유사성 측도(similarity metrics), 수정규칙(modification rules), 교정규칙(repair rules) 등의 지식을 이용하게 된다(Gupta, 1994; Slade, 1991). <그림 2>는 4 REs와 이상의 다섯 가지 지식들을 이용해서 새로운 문제를 해결하는 CBR 과정모형으로서 이는 문제발생시마다 반복적으로 발생하는 순환시스템의 성격을 가진다. 순환시스템의 첫 번째 단계는 새로운 문제를 새로운 사례로 인식하여 이의 주요 속성에 대한 색인을 이용해서 사례베이스를 검색하는 단계이다. 두번째 단계는 사례베이스의 검색결과와 새로운 사례를 비교하는 단계로서 이때 유사성 측도를 이용해서 검색된 사례들중 가장 유사성 측도가 높은 사례를 기저사례로 선택하며, 선택된 기저사례는 문제해결을 위해 재사용된다. 세번째 단계는 기저사례로 선택된 기존 사례가 새로운 문제에 대한 사례와 완전히 일치하지 않을 경우 수정규칙을 이용해서 기존사례의 내용을 수정해서 새로운 사례에 적합한 해결안을 생성하는 단계이다. 네번째 단계는 새로운 사례가 기존의 사례와 상이하였을 경우 교정규칙을 이용해서 사례베이스에 새로운 사례를 저장하는 단계이다.

3. 개별사례의 구성요소 및 표현요구사항

사례내에 정확히 어떤 정보가 포함되어야 하는가에 대해서는 아직 합의가 이루어지지 않은 상태이다. 그 이유는 첫째, CBR에 대한 연구동기가 2가지 방향, 즉 인간의 지식집약적 업무수행을 모형화하려는 의도와 기존의 데이터베이스나 지식저장소를 이용하려는 의도에서 나왔기 때문이다. 둘째는 CBR의 다양한 업무영역이 사례표현에 대해 다양한 기능요구사항을 부과하기 때문이다(Alterman, 1989). 그러나 일반적으로 사례는 문제기술 부분과 해결안 부분을 주요 구성요소로 갖는다(Mahapatra 등, 1994). 문제기술 부분은 의사결정과 관련된 문제의 속성에 대한 정보를 가지며, 해결안 부분은 의사결정이나 문제해결에 필요한 해결절차에 대한 정보를 가지고 있다.

문제기술 부분의 내용은 사례의 미래 용도와 새로운 문제해결 상황과의 비교를 위해 어떤 정보가 필요한가에 따라 다르다. 어떤 기술적 정보(descriptive information)가 문제기술 부분에 포함되어야 할 것인가에 대한 결정은 <표 1>의 두가지 일반규칙을 따르면 된다(Kolodner, 1993).

특히 기술적 정보의 포함 여부에 대한 결정규칙은 이론적 기반이 취약한 영역에 있어서 중요시 되며, 규칙 1은 명확히 관련된 것들은 기록되어야 한다는 것을 의미한다. 따라서 관련은 되나 변동이 전혀 없는 암묵적인 속성들에 대해서는 시스템이 무시할 수 있다는 것이다. 규칙 2는 접근가능성을 위해 필요한 규칙이다. 일반적으로 과거의 사례에 대한 기술내용이 새로운 문제에 대한 기술내용과 유사할수록 검색과 비교는 더욱 쉬워진다. 따라서 새로운 문제를 해결하기 위한 정보나 새로운 상황을 해석하기 위한 정보는 문제기술 부분에 포함되어야 한다.

해결안 부분의 내용은 시스템의 응용분야에 따라서 매우 다양한 형태를 갖는다. 설계 시스템에 있어서 해결안 부분은 설계한 인공조형물이 되며, 계획 시스템에 있어서 해결안 부분은 파생된 계획이 된다. 또한 해석 시스템에 있어서 해결안 부분은 해석이나 분류 결과가 되며, 설명 시스템에 있어서는 설명 그 자체가 해결안이 된다. 따라서 각 사례에 기술되어 있는 해결안 부분은 문제기술 부분에 기술되어 있는 목표의 달성이나 제약조건을 충족시켜 줄 수 있는 개념이나 대상들이라 볼

규칙 1	목표달성을 위해 명확히 고려 대상이 되는 모든 정보를 포함해야 한다.
규칙 2	일반적으로 해당 유형의 사례를 기술할 때 이용되는 정보를 포함해야 한다.

<표 1> 문제기술 부분에 포함될 정보의 결정규칙

수 있다.

CB에 많은 수의 사례가 저장되어 있어도 동일한 상황에 대한 사례라고 한다면 각 사례의 상대적 가치는 하락할 것이다. 따라서 CBRS의 문제 해결능력은 저장된 사례의 수와 다양성에 의존한다(Mahapatra 등, 1994; O'Leary, 1993). 그러므로 사례 저장문제는 동일한 사례는 배제하고 유일한 사례와 우위 사례(superset case)만을 저장한다는 사례저장 휴리스틱에 따라야 한다(Liang, 1993).

따라서 사례 표현 스킴은 이러한 기준에 따라 효율적인 지식표현구조가 되기 위해서는 Fedorowicz 등(1986)이 밝힌 일반적으로 우수한 지식표현구조가 갖는 특성인 ① 응용 도메인에 대한 추상 제공 ② 표현의 완전성과 간결성 ③ 효율적 접근가능성 ④ 문제의 중요사항은 강조하고 완전성을 위해 필요하나 이용빈도가 낮은 세부사항은 은폐할 수 있는 특성 이외에 다음과 같은 특성을 추가로 필요로 한다.

(1) 사례의 유일성 : 동일한 사례를 중복 저장하는 것은 CBRS의 문제해결능력의 향상은 없으나 저장공간의 낭비를 초래하게 된다. 따라서 사례 표현 스킴은 사례의 유일성을 보장함으로서 사례의 중복 저장을 배제시켜 기억공간을 절약함과 동시에 효율적 접근가능성을 향상시킬 수 있어야 한다.

(2) 해결안과 의사결정 목표의 독립성 : 동일한 해결안으로 다양한 상황에서 다양한 의사결정 목적을 달성할 수 있다(Liang, 1993). 그러므로 각 사례는 과거의 이용목적과 관계없이 유사한 특성을 가진 새로운 목적을 위해 이용 가능하여야 한다. 따라서 해결안과 의사결정 목표의 독립성을 보장하기 위해 사례표현시 목적을 하나의 속성으로 취급하여야 한다.

(3) 사례표현의 유연성 : 사례는 유사성 평가를 통해 활용된다. 유사성 개념에는 개체 유사성(entity similarity), 구조적 유사성(structural similarity) 및 기능적 유사성(functional similarity) 등이 있다(Liang 등, 1993). 따라서 사례는 이들 유사성 개념을 지원할 수 있도록 표현되어야 한다.

(4) 사례의 복합성 : 일반적으로 사례는 개별적으로 고립된 것이 아니라 상호작용을 하게 된다. 따라서 사례는 동일한 차원에 따라 다른 사례와 연결될 수 있다. 사례는 사상(event)을 표현한 것으로서 개별 사상이 다른 큰 사상의 일부가 될 수 있는 것과 마찬가지로 사례도 다른 큰 사례의 일부가 될 수 있다. 또한 사례는 보다 작은 사례를 자신의 일부로 가질 수 있다. 따라서 대형 사례는 인과적이거나 순차적 혹은 단편적으로 연결된 사례들을 결합한 것이 된다.

Ⅲ. CAPP를 위한 CB의 계층적 구조화

1. CAPP 응용분야

컴퓨터지원 공정계획(Computer Aided Process Planning : CAPP)은 CBR의 특성을 이용할 수 있는 적합한 응용 도메인중 하나이다. 공정계획이란 공학적 도면의 정의에 따라 부품을 초기 형태에서 최종 형태로 전환하는데 필요한 제조공정과 매개변수의 정의에 관련된 것이다(Rogers 등, 1994). CAPP란 컴퓨터의 지원을 받아 공정계획수립을 자동화하는 분야로서 CAD(Computer Aided Design)와 CAM(Computer Aided Manufacturing)의 교량 역할을 한다. CAPP를 위한 접근법으로는 변성형 접근법(variant approach)과 생성형 접근법(generative approach)으로 2대별 될 수 있다(Irani 등, 1995).

변성형 접근법은 새로운 부품에 대한 공정계획이 어떤 부품계열의 표준계획을 수정함으로서 수립될 수 있다는 가정을 하고 있다. 따라서 새로운 부품에 대한 공정계획은 유사한 부품에 대한 기존 계획을 검색한 후 새로운 부품에 적합하게 필요한 수정을 하므로서 생성된다. 일부 변성형 시스템들은 제조방법의 유사성에 따라 부품들을 다수의 부품계열로 집단화하기도 한다. 변성형 시스템은 정보

관리 능력을 증대시켜 줌으로서 공정계획에 관련된 활동과 의사결정의 노력 및 시간을 줄여 줄 수 있다. 변성형 접근법이 대중적인 이유는 투자비용이 적고, 개발시간이 짧다는 것과 개발비용과 H/W 비용이 적게 소요된다는 것이다. 그러나 변성형 시스템은 제조방법의 수정시 일관성 유지가 어려우며, 기하학적 형태, 크기, 정밀도, 자재, 품질, 공정부하 등의 다양한 결합을 수용하기 어렵고, 공정계획의 질이 공정계획자의 지식배경에 의존한다는 단점이 있다. 변성형 접근법을 이용한 시스템으로는 CAPP, MIPLAN 등이 있다(Alting 등, 1989).

생성형 접근법은 CAD와 부품 형상정보로부터 획득 가능한 정보를 입력 자료로 해서 부품에 대한 공정계획을 합성하려는 접근법이다. 제조규칙과 설비능력을 컴퓨터 시스템에 저장해 두고 공정계획자의 개입없이 특정 부품에 대한 공정계획을 생성하게 된다. 생성형 접근법의 장점은 공정계획의 일관성과 완전 자동화에 있다. 따라서 종전에 제작한 경험이 없는 새로운 부품에 대해서도 생산공정에 관한 원칙이나 원리에 기초해서 자동적으로 공정계획을 생성할 수 있다. 그러나 생성형 접근법은 복잡성과 개발의 어려움으로 인해 개발시간과 비용이 많이 소요된다. 생성형 접근법을 이용한 시스템으로는 APPAS, CMPP, EXCAP, XPLAN 등이 있다.

CBR 기법은 과거의 문제해결 경험을 새로운 문제해결에 이용하려는 방법이다. 따라서 CAPP에 대한 접근법 관점에서 보면 CBR 기법은 변성형 접근법에 해당된다. 본 논문에서는 변성형 접근법으로서의 CBR 기법의 기능을 강화시키기 위하여 전통적 CBR 기법에서 일반적으로 이용하는 기존의 문제해결 경험은 DDCB에 저장해두며, 문제해결 경험이 없는 새로운 문제해결을 위하여 일반적 지식은 DICB에 저장하므로서 공정계획수립의 가능성을 증대시킬 것이다. 일반지식을 지식베이스로 구축하지 않고 CB 형태로 구축하는 이유는 추론메커니즘을 추가로 도입하지 않기 위해서이다. 만약 일반지식을 지식베이스에 저장할 경우 이들의 표현 형식은 생성규칙의 형식이 될 것이므로 관련 지식을 검색하기 위해서는 RBS에서 이용되는 또 다른 추론메커니즘이 필요하게 된다. 따라서 일반지식을 사례형식으로 표현하여 별도의 CB를 도입하는 것이 구축의 측면에서 볼 때 합리적이라 할 수 있다.

2. 계획수립문제의 특성

사례를 이용한 계획수립을 사례기반형 계획(Case-Based Planning : CBP)이라고 하며 CBP 시스템의 입력요소는 대상 객체의 초기 상태이며, 결과는 계획이 된다. 계획은 대상 객체를 원하는 상태로 변환시키는 행위절차들이다. 계획수립은 근본적으로 탐색문제이다(Hendler 등, 1990). Gupta 등(1990)은 계획수립문제의 계산복잡성이 NP-hard함을 증명하였다. 따라서 계획수립문제는 해결안공간을 구조화시키는 방법에 따라 검색시간에 많은 차이가 발생한다. CB는 계획을 사례형식으로 저장해둘 곳이다. 만약 CB에 저장된 사례들의 수가 적을 경우에는 플랫구조(flat structure)를 이용해서 CB를 구조화해도 저장공간과 반응속도에 큰 영향을 미치지 않을 것이다. 그러나 사례의 수가 많을 경우에는 사례를 단순 리스트나 배열의 형식으로 저장하는 플랫구조형식에서는 사례 저장을 위한 저장공간 요구량이 급증할 뿐만 아니라 순차적 방법으로 관련된 사례들을 검색하기 때문에 반응속도가 사례수의 증가에 따라 급격히 늘어날 것이다.

추상화는 계산의 복잡성을 효과적으로 줄여줄 수 있는 방법이다. 추상화의 기본적 사고는 문제해결의 초기 단계에서는 문제의 기본적 특성들만을 고려한후 하위수준의 세부적인 사항들은 추후에 충족시킨다는 것이다. 추상화기법을 이용하므로서 해결안공간은 계층적 구조로 표현된다. 추상화 계층구조는 기저공간(basic space)과 추상화공간(abstract space)으로 구성된다. 만약 n개의 기저공간을 갖는 문제를 m 단계의 추상화계층구조로 표현할 경우 이를 대상으로 한 계획수립시간은 $\sum_{i=1}^{m} 2k_i$ (k_i : i 번째 추상화공간과 기저공간간의 비율)가 된다(Korf, 1987). 따라서 추상화 기법은 계산시간의 복잡성 함수를 지수함

수에서 선형함수로 바꾸어 준다. 따라서 계획수립문제의 한 형태인 CAPP의 경우에도 공정계획에 대한 사례들을 추상화된 계층적 구조로 조직화하는 것이 효율적인 것이다.

문제공간을 계층적 구조로 추상화시킬 경우 기저공간에 표시되는 지식은 세부사항들이 되며, 상위수준의 추상화공간에 표시되는 지식은 기저공간에 표시된 객체들이 공통적으로 갖는 주요 속성들이 된다. 따라서 계획사례들을 갖는 CB의 구조는 <그림 3>과 같은 형태가 될 것이다.

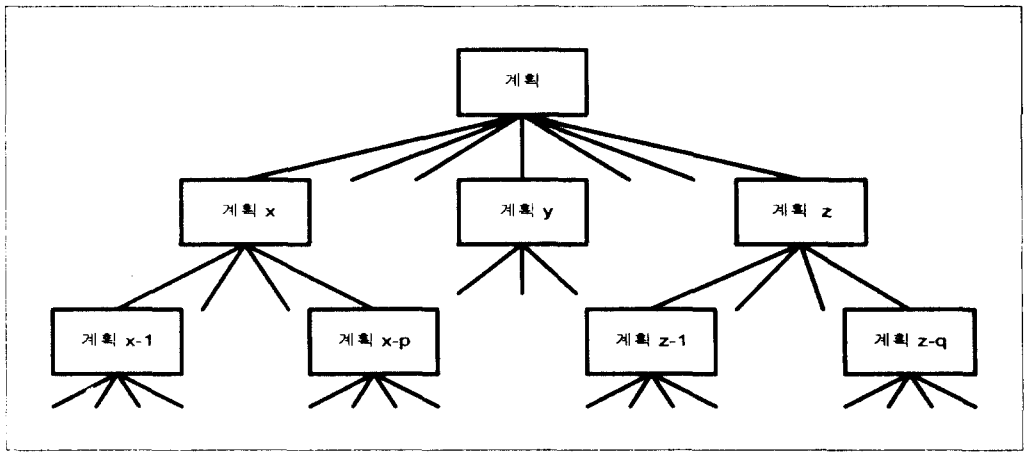
사례의 추상화는 관련 사례들의 유사성측면에서 상향식으로 이루어진다. 유사성개념에는 세가지 측면이 있다. 첫째는 개체 유사성으로 이는 두 개체가 지닌 속성들을 비교해서 측정하는 것으로, 이는 의미론적인 측면에서의 유사성 개념이라 할 수 있다. 둘째는 구조적 유사성으로 대응하는 두 개체의 내부관계를 비교해서 측정하는 것으로 구조적 동질성 측면에서의 유사성 개념이라 할 수 있다. 세째는 기능적 유사성으로 두 개체가 유사한 기능을 발휘하거나 유사한 목표를 달성하는가에 따라 측정되는 유사성 개념이다. 이는 실용적인 측면에서의 유사성 개념이라 할 수 있다. 따라서 사례를 어느 유사성관점에서 보는가에 따라 계층적 구조가 달라질 것이다. 일반적으로 사례를 유사성에 의해 계층화할 경우에는 표면적인 속성을 주로 이용하므로 개체유사성에 따른 계층적 구조형태가 될 것이다.

CB가 계획에 관한 사례를 계층화하므로서 얻을 수 있는 이점은 계획수립시간의 절감뿐 아니라 새로운 사례의 추가 및 CB의 수정을 용이하게 할 수 있다는 것이다(Paek 등, 1996). 또한 CB를 계층화할 때 상속성개념을 이용할 경우 저장공간을 절약할 수 있으며, 자료의 일치성도 증대될 것이다.

3. DDCB의 구조화

1) 사례베이스의 계층적 구조화

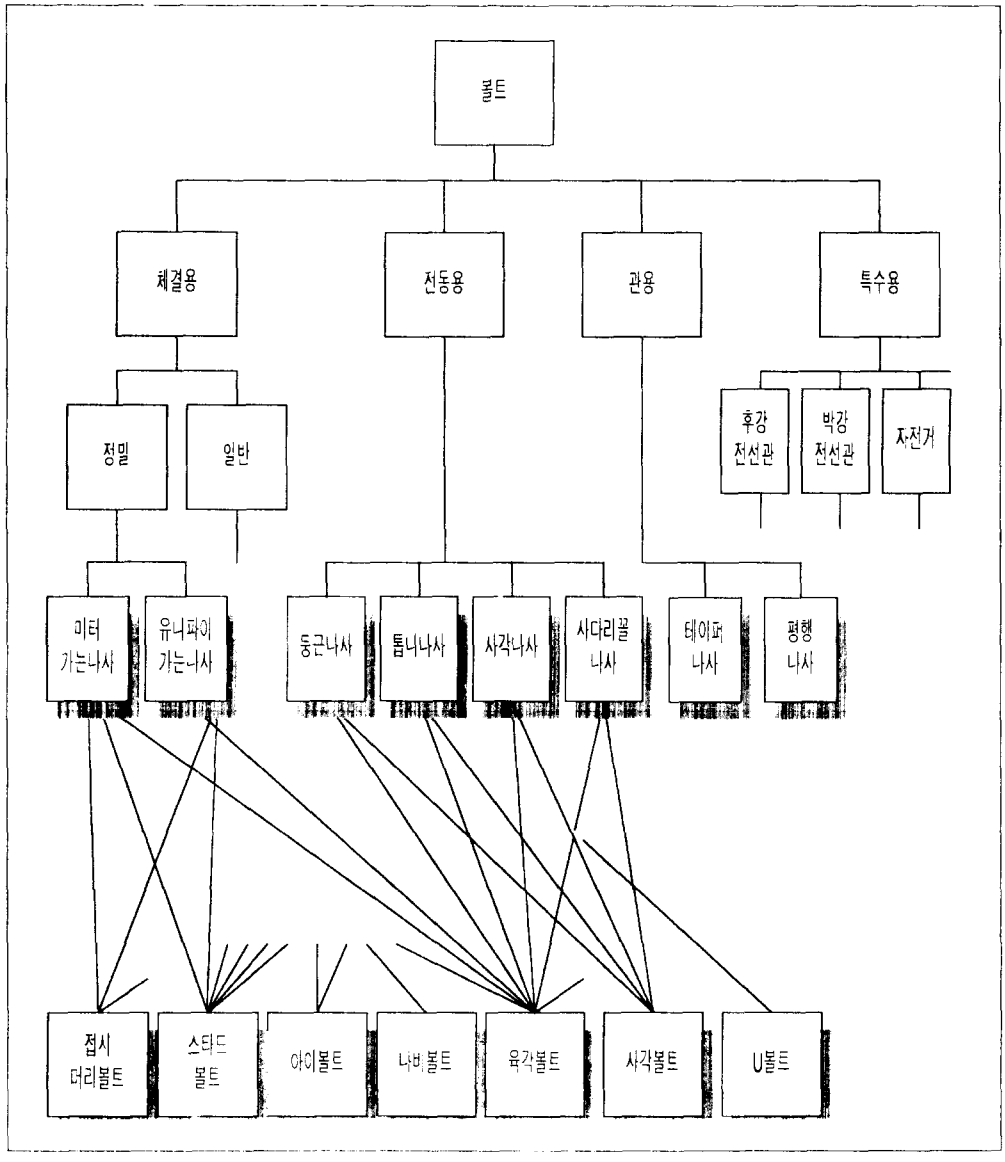
본 논문에서는 볼트의 가공공정 결정문제를 CBR 기법에 적용시켜 보기로 한다. 볼트의 부품 형태는 비교적 간단하나 그 종류는 머리모양, 나사산의 모양 등에 따라 매우 다양화되어 있다. 그러나 이러한 볼트의 다양화는 용도, 나사산의 모양, 머리모양 등에 있어서 공통적인 요소가 많이 존재한다. 볼트는 용도에 따라 체결용, 전동용, 관용, 특수용으로 세분할 수 있으며, 체결용 볼트는 사용분야에 따라 정밀용과 일반용 볼트로 나눌 수 있다. 정밀용 볼트는 나사산의 모양에 따라 미터 가는 나사와 유니파이 가는나사로 세분될 수 있으며, 이들은 머리모양에 따라 다양한 형태의 볼트로 보다 세분화될 수 있다. 일반용 볼트는 나사산의 모양에 따라 미터보통나사, 미니츄어나사, 유니파이 보통나사 등으로 세분될 수 있으며, 이들 또한 머리모양에 따라 다양한 형태의 볼트로 보다 세분화될 수 있다. 전동용 볼트는 나사산의 모양에 따라 둥근나사, 톱니나



<그림 3> 계층적 계획 CB 구조

사, 사각나사, 사다리꼴나사 등으로 세분될 수 있으며, 이들 모두 육각 혹은 사각볼트가 될 수 있다. 또한 관용 볼트는 나사의 형태에 따라 테이퍼형과 평행형으로 나뉘며, 특히 관용 볼트는 머리부분이 없는 것이 특징이다. 특수용은 사용용도가 매우 특수해서 계층화가 어려우나 후강 전선관용, 박강 전선관용, 자전거용 등으로 나눌 수 있다.

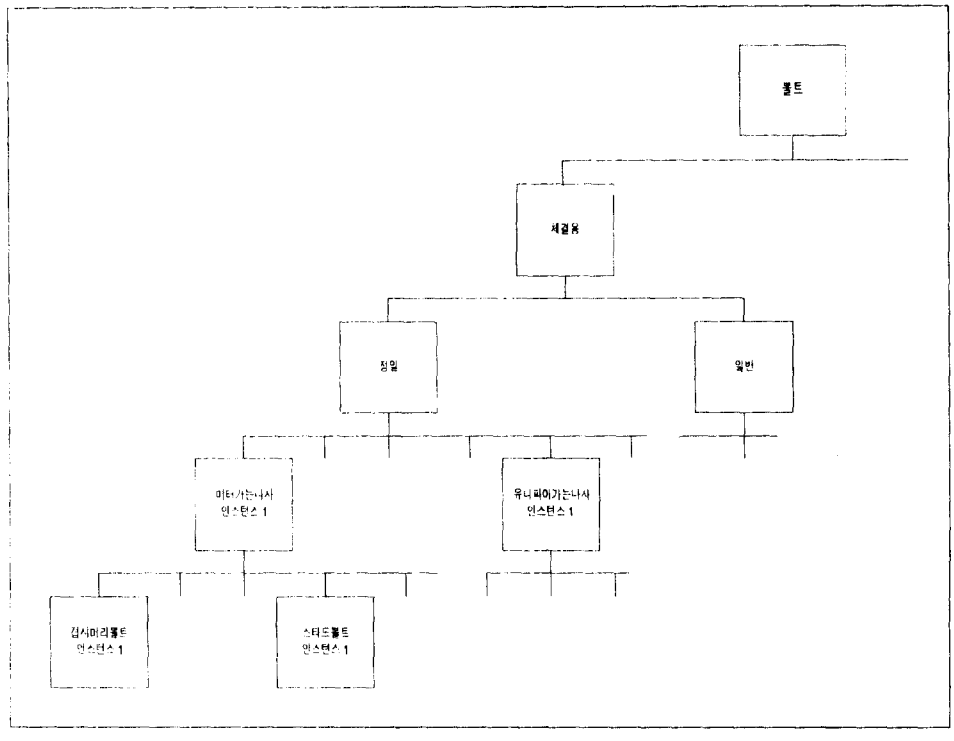
이상에서 살펴본 바와 같이 볼트는 간단한 형태이기는 하나 종류는 나사부분과 머리부분의 결합형태에 따라 매우 다양함을 알 수 있다. 따라서 이들 볼트를 각각의 유형별로 사례화할 경우에는 중복저장문제가 심각해질 것이므로 <그림 4>와 같이 볼트구조에 따라 사례를 계층화시킨 후 하위수준의 사례들이 상위수준의 사례정보를 상속받게 하는 것이 합리적일 것이다. 사례들간의 관계측면에서 볼 때 볼트의 계층구조상에서 루트로부터 나사산의 모양에 따른 분류까지는 1 : n의 isa 관계형태로 상위사례와 하위사례가 연결되어 있다. 머리모양에 따른 분류단계는 나사산 모양에 따른 상위사례와 n : n의 관계로서 상/하위사례가 연결되어 있다. 따라서 이들 n : n의 관계에 관련된 머리모양의 사례들은 중복저장을 해야 한다. 사례의 중복저장에 따라



<그림 4> 볼트의 계층적 구조

저장공간의 증가로 인해 비용이 증가하나 사례구조에 대한 이해도 향상과 검색비용 감소로 인한 효용이 보다 클 것이다. 특히 머리모양에 따른 개별사례의 문제기술부분에는 머리부분에 대한 정보뿐만 아니라 나사산부분에 대한 정보도 가지고 있다. 따라서 이는 상위수준이 하위수준에 대해 part-of의 관계를 갖는 경우이다. 이 경우는 사례표현의 효율성을 위해 상속개념을 이용해서 상위수준 사례의 사례식별자를 하위수준 사례에 표시해야 한다.

또한 개별사례 클래스의 구성요소 수의 측면에서 볼 때 루트에서부터 용도에 따른 분류단계까지는 사례 클래스들이 1개의 사례만을 가지나 나사산의 모양에 따른 분류단계 이하의 사례 클래스들은 다수의 사례들을 구성요소로 가진다. 따라서 DDCB의 구조는 <그림 5>와 같은 계층적 구조형태를 가지게 된다.



<그림 5> DDCB의 구조

2) 개별사례의 표현예

사례들은 한정된 수의 알려져 있는 단어(vocabulary)나 수치값을 갖는 속성들의 쌍(pair)인 튜플형식으로 표현되는 것이 일반적이다. 그러나 이러한 표현 형식은 part-of의 관계를 갖는 복합사례나 대형사례의 표현이 어렵다. 그러므로 이러한 문제를 해결하기 위해서는 사례를 구조적 표현법(structured representation)에 따라 표현해 주는 것이 좋다. 구조적 표현법의 이점은 2가지로 요약될 수 있다(Plaza, 1995). 첫째는 복합객체(composite objects)의 표현이 자연스럽다는 것이다. 둘째는 구조적으로 표현된 사례는 사례의 일부분을 완전한 사례로 취급할 수 있는 능력을 제공해준다는 것이다. 따라서 사례의 일부분이 새로운 문제의 일부 혹은 전부를 해결하는데 이용될 수 있으며, 또한 검색된 여러 사례를 이용해서 하나의 문제를 해결할 수 있게 된다.

사례의 문제 혹은 상황 기술부분은 상태를 코드화한 부분이다. CBR 과정에서 문제기술부분은 새로운 문제와 기존 사례간의 유사성 평가를 위해 이용된다. DDCB를 이용해서 CAPP 업무를 지원한다는 것은 과거의 관

(가) 상속개념 적용전의 사례표현

사례식별자 : 접시머리볼트 인스턴스 1-1;
용　　도 : 정밀;
사례사용일 : 97년 5월 24일;
이용 빈도 : 12회;
머리 속성 : 종　　류 : 접시;
　　상부 직경 : 20;
　　높　　이 : 5.5;
　　홈 모 양 : +;
　　변화 각도 : 90도;
나사 속성 : 종　　류 : 미터가는;
　　호　　칭 : M10;
　　피　　치 : 1.25;
　　접촉 높이 : 0.677;
　　외　　경 : 10;
　　몸통 길이 : 50;
　　나사부길이 : 35;
　　끝 모 양 : 둥근끝;
공정 계획 : 1. 절단
　　2. 모서리면취
　　3. 후렉션기계가공
　　4. 스크랩제거
　　5. 전조작업
　　6. 머리홈생성
　　7. 열처리
　　8. 도장

(나) 상속개념 적용후의 사례표현

사례식별자 : 접시머리볼트 인스턴스 1-1;
사례사용일 : 97년 5월 24일;
이용 빈도 : 12회;
머리 속성 : 종　　류 : 접시;
　　상부 직경 : 20;
　　높　　이 : 5.5;
　　홈 모 양 : +;
　　변화 각도 : 90도;
나사 속성 : 미터가는나사 인스턴스 1-6;
공정 계획 : 1. 절단
　　2. 모서리면취
　　3. 후렉션기계가공
　　4. 스크랩제거
　　5. 머리홈생성

<그림 6> 상속개념 적용전후의 사례표현 예

련 사례를 선택한 후 사용자의 요구사항에 적합하도록 과거 사례를 수정해서 공정계획을 수립하는 것을 의미한다. 이때 DDCB에서의 관련 사례검색은 볼트의 속성에 대한 사용자의 문제입력에 따라 이루어지며, 사용자의 문제입력내용은 사례의 구성요소중 문제기술부분에 해당된다. 문제기술부분은 볼트의 용도, 머리부분 및 몸통부분 등에 대한 정보로 구성될 수 있다. 만약 새로운 문제 혹은 상황이 기존 사례들중 어느 것과 문제기술부분이 충분히 유사할 경우 해당 사례를 선택해서 문제의 해결을 시도한다. 따라서 문제기술부분은 새로운 상황이나 문제에 기존 사례의 적용가능성을 평가할 수 있도록 충분히 세부적이라야 한다.

예를 들면 DDCB의 가장 하위계층에 위치하는 접시머리볼트의 경우 이의 문제기술부분이란 가공에 필요한 볼트의 속성부분을 의미하며, 이는 크게 머리부분의 속성과 나사부분의 속성으로 분류될 수 있다. <그림 6>은 접시머리볼트에 대한 사례표현 예이다.

<그림 6>의 (가) 부분은 상속개념이 적용되기 전의 사례표현 예로서 문제기술부분에 있어서 용도와 나사의 속성부분은 상위단계에서 이미 기술된 정보와 동일하다. 즉, 나사부분의 속성 및 공정계획에 대한 정보는 상위단계의 사례에도 기술되어 있어 저장공간의 낭비를 초래하므로 상속개념을 이용해서 나사사례에 대한 사례식별자(case identifier)만을 표시하는 것이 효율적일 것이다. 따라서 (가) 형태의 사례표현은 (나) 형태로 수정되어야 한다. 그리고 용도는 사례가 나타내는 객체의

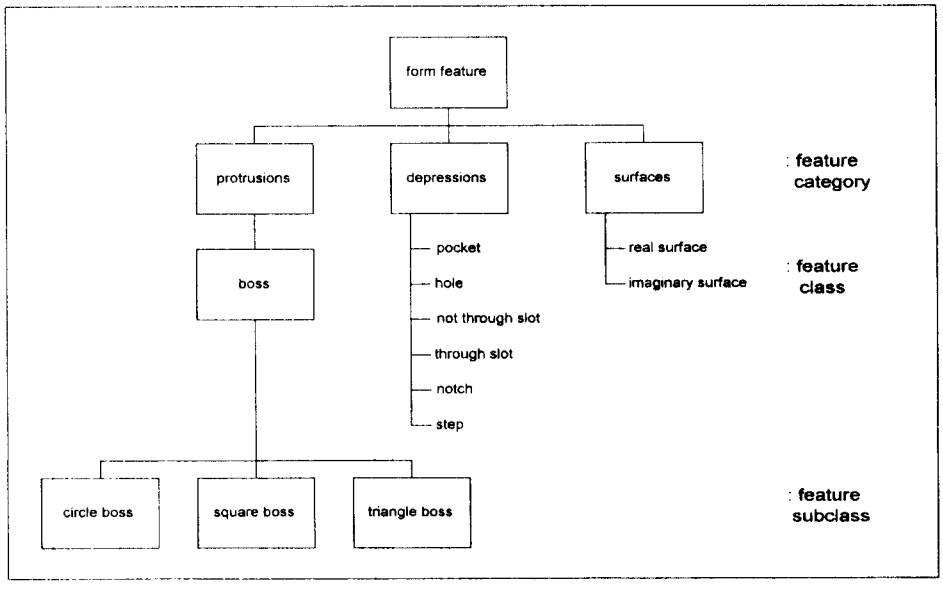
이용목적을 의미하는 것으로 이는 나사사례의 상위단계인 용도사례에 나타나 있다. 용도를 속성으로 처리함에 따라 해당 사례를 동일한 목적인 경우 다양한 분해상황에서 이용가능하게 할 수 있을 것이다. 공정계획부분은 해당 사례의 과거 해결안을 의미한다.

3. DICB의 구조화

1) 사례베이스의 계층적 구조화

공정계획의 전문가는 부품의 생산계획에 관한 전문지식을 가지고 있다. 전문가가 가지고 있는 전문지식중 일반적인 공정계획지식은 부품의 세부형태와 독립적으로 사례형식으로 저장되어질 수 있다. 공정계획 정보가 CAD와 CAM에서 교량 역할을 수행하기 위해서는 두 시스템들과의 의사소통 매체로서 특성(feature)을 이용하게 된다. 특성이란 부품의 형태 혹은 기타 특성의 제측면에 관한 정보집합으로 볼 수 있다(Salomons 등). 따라서 특성은 부품의 기하학적 형태를 인식 가능하고 의미있는 형태로 분할함으로서 부품의 특징에 상위수준의 개념적 의미를 부여해 준다(Gindy, 1989).

특성 유형에는 형태특성(form feature), 기능특성(functional feature), 조립특성(assembly feature), 물리적 특성(physical feature) 등이 있으며, 본 논문에서는 형태특성을 대상으로 논할 것이다. 형태특성은 진입경계(entry boundary), 진출경계(exit boundary), 깊이경계(depth boundary)로 구성된다. 진입경계는 가공을 위해 기계나 장비가 접근하는 실제 혹은 가상의 면, 선 혹은 점으로서 특성에 따라 여러 개가 존재할 수 있다. 진출경계는 가공이 완료되는 부분으로서 진입경계의 반대편에 존재한다. 깊이경계는 가공시 기계나 장비가 진입경계에서 진출경계로 이동시 형태의 진행을 나타내는 것으로 깊이축(depth axis)을 따라 특성 볼륨을 둘러싸는 면들의 집합이다. 특성의 진입/출경계는 실제 부분(solid part)의 선(edge)으로서 연속적인 폐쇄루프 형태로 주변형태(perimeter)를 표현 가능하면 폐쇄(closed)라고 하며, 그 이외의 경우는 개방(open)이라고 한다. 특성 기술시 깊이경계에 대해 고려할 사항은 깊이축의 유형 및 대칭성, 진입/출 주변의 기하학적 형태(perimeter geometry)간의 관계, 그리고 깊이축을 따라 가면서 형태가 변하는 유형 등이다. 이들 4가지 매개변수의 값에 따라 깊이경계는 평행형(constant), 테이퍼형(tapered), 볼록형(convex), 오목형(concave), 등고형(contoured) 등의 형태로 분류 가능하다. 진출경계가 가상인 경우에는 관통(through) 상태라고 말한다.



<그림 7> 형태특성의 계층구조

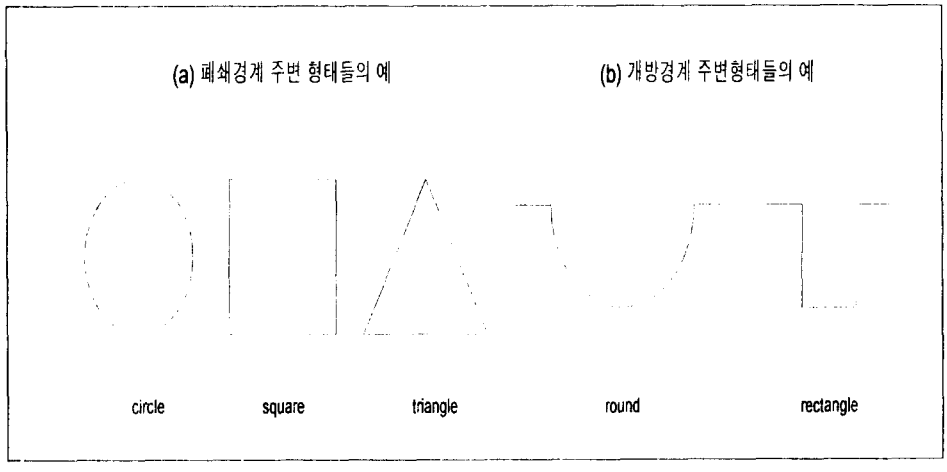
내부특성(internal feature)의 경우 진출경계가 관통인 상태는 특성이 바닥표면(bottom surface)을 갖지 않는 경우를 의미한다. 외부특성(external feature)의 경우 관통인 상태는 진입경계의 반대편에 있는 진출경계가 가공시 기계나 공구에 의해 접근 가능함을 의미한다.

위와 같은 기준에 따라서 형태특성을 계층적으로 분류하면 <그림 7>과 같다. 특성분류(taxonomy)의 이점은 구조적 방법으로 특성을 분류하는데 이용할 수 있으며 특성 인식 시간이 절감된다는 것과 상속 개념으로 인해 간결한 지식 표현이 가능하다는 것이다. <그림 7>에서 protrusions, depressions, surfaces는 특성 카테고리(feature category)를 의미하며, boss, pocket, hole 등은 기계나 공구가 가공시 접근 가능한 방향의 수, 진입/출경계의 개폐, 진출경계의 실재 여부 등의 기준에 따라 특성 카테고리를 상세화한 특성 클래스(feature class)에 해당한다. 이들 특성 클래스들은 공학적 응용에 일반적으로 사용되는 인식 가능한 기하학적 형태로서 특성 카테고리들을 세분화한 결과이다.

특성 클래스들은 주변의 기하학적 형태와 결합하여 특성 서브클래스(feature subclass)로 세분할 수 있다. 특성을 서브클래스 단계까지 분류하면 완전히 인식 가능한 개체가 될 수 있다. <그림 8>은 일부 주변형태들의 예이다. 주변의 기하학적 형태는 특성 클래스와 밀접한 관계를 가진다. 즉, boss, hole, pocket은 외부접근방향(External Access Direction : EAD)의 범위가 "0 ≤ EAD ≤ 2" 이므로 폐쇄경계 기하학적 형태와 결합이 가능하다. 또한 slot은 EAD의 범위가 "2 ≤ EAD ≤ 3" 이므로 2 혹은 3개의 EAD를 갖는 개방경계 기하학적 형태와 결합될 수 있으며, notch와 step의 EAD의 범위는 "3 ≤ EAD ≤ 4" 이므로 3 혹은 4개의 EAD를 갖는 개방경계 기하학적 형태와 결합될 수 있다. 그리고 실제 및 가상 표면(real/imaginary surface)은 EAD의 범위가 "5 ≤ EAD ≤ 6" 이므로 5 혹은 6개의 EAD를 갖는 개방/폐쇄경계 기하학적 형태와 결합될 수 있다.

만약 <그림 8>에 있는 폐쇄경계 주변형태와 boss가 결합되면 boss는 circle boss, square boss, triangle boss 등의 특성 서브클래스로 세분될 수 있다. 현실적인 필요에 따라 특성 서브클래스들은 보다 세분화될 수 있으나 그 결과 나타나는 기하학적 형태들은 독립적으로 존재하는 특성이 될 수 없으므로 기본 특성 위에 놓여지는 부분적인 기하학적 형태가 된다. 따라서 본 논문에서의 특성 분류는 특성 서브클래스 수준까지로 한정하기로 한다.

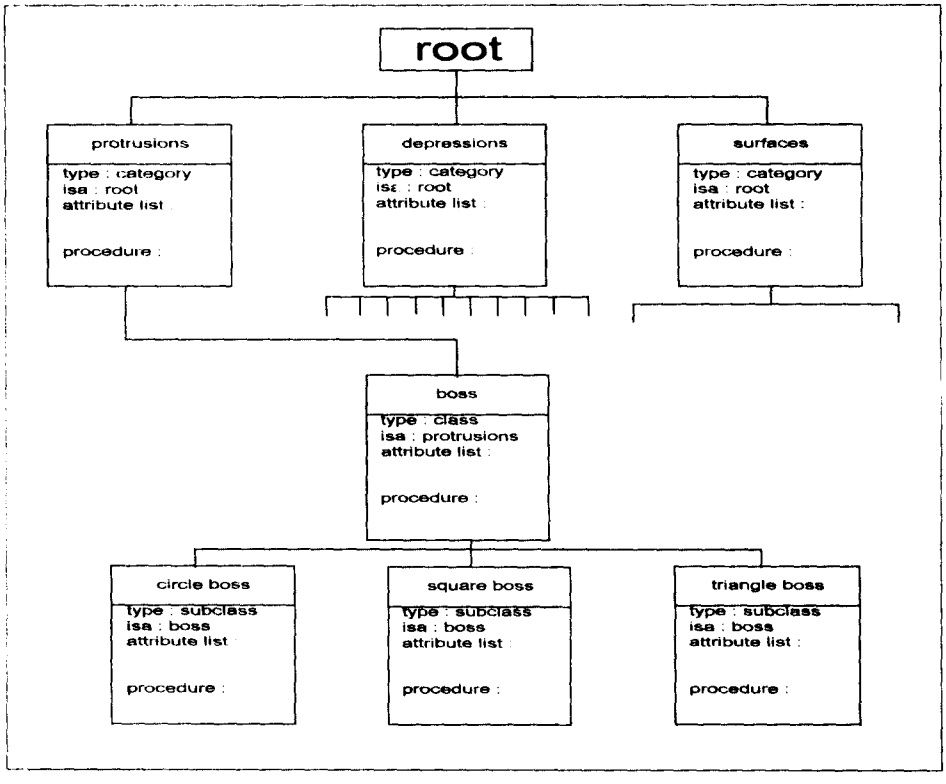
위에서 분류한 형태특성들은 사례형식으로 표현되어 DICB에 저장된다. 따라서 DICB는 형태특성에 관한 정보를 사례형식으로 가지고 있는 부분이다. 형태특성들이 <그림 7>과 같이 계층적 구조를 가지므로 DICB도 계층적 구조로 조직화하므로서 자료표현의 일관성을 유지할 수 있다. 따라서 DICB의 구조는



<그림 8> 주변형태의 예

<그림 9>와 같은 형태가 된다.

이때 각 특성 클래스와 서브클래스들의 내용을 비교함으로 중복된 내용은 상위수준의 특성 카테고리와 클래스에 저장하고 특성 서브



<그림 9> DICB의 구조

2) 개별사례의 표현예

DICB는 전문가가 지닌 일반적 지식을 사례형식으로 저장해 놓은 부분이다. 따라서 DICB에 저장되는 사례의 수는 DDCB에 비해 상대적으로 적다고 할 수 있다. 그러나 DICB에 저장된 각 사례들은 DDCB에 저장된 사례에 비해 상대적으로 이질성이 크다고 할 수 있다. 따라서 사례의 수는 적으나 대부분의 사례들이 유일성이 높으므로 DICB의 문제해결능력은 사례의 수에 비해 크다고 할 수 있다.

앞에서 살펴본 바와 같이 DICB는 계층적 구조로 되어 있으며 상하위 수준간의 관계는 모두 isa의 관계로 연결되어 있다. 따라서 DICB내의 사례들은 상하위수준간에 공유정보의 양이 많아 상속개념을 도입하므로서 사례표현의 효율성을 증대시킬 수 있을 것이다.

모든 특성 카테고리, 클래스와 서브클래스들은 DICB내에 개별적인 사례로 저장된다. 이때 각 특성 클래스와 서브클래스들의 내용을 비교함으로 중복된 내용은 상위수준의 특성 카테고리와 클래스에 저장하고 특성 서브클래스들이 갖는 상세 정보만을 특성 서브클래스의 사례에 표현해도 정보표현의 완전성 기준을 위배하는 것이 아니다. 따라서 사례들을 계층형으로 조직화할 경우 서브클래스들의 문제기술부분은 <그림 10>의 (가) 형태보다는 (나) 형태로 표현되어야 한다.

<그림 10>의 (나)는 상속개념이 적용된 후의 circle boss에 대한 사례표현 예이다. 상속개념이 적용되므로서 문제기술부분의 속성들의 수가 많이 감소하였음을 알 수 있다. 이들 감소된 속성들은 모두 상위수준의 boss 사례에 표현된 속성들이다. 즉, 상속개념에 의해 boss 사례의 속성들은 모두 circle boss 사례가 상속받을 수 있으므로 이들을 표시하지 않는다. boss 사례의 속성들을 circle boss 사례가 상속 가능하도록 해주는 부분이 isa 속성이다. 즉, DICB는 부모와 자식간의 관계가 1 : n의 형태만을 가지므로 isa의 관계만으로 모든 사례들이 연결가능하다. 따라서 isa 속성

(가) 상속개념 적용전의 사례표현

```
case identifier : circle boss;
type : subclass;
isa : boss;
attribute list : height : positive;
                 I/E feature : external;
                 entry surface : real;
                 exit surface : real;
                 boundary : closed;
                 exit status : not through;
                 EAD's # : 0;
                 perimeter geometry : circle;
```

(나) 상속개념 적용후의 사례표현

```
case identifier : circle boss;
type : subclass;
isa : boss;
attribute list : perimeter geometry : circle;
```

<그림 10> 상속개념 적용전후의 사례표현 예

은 DICB의 계층구조 특성을 활용하는 부분이라 할 수 있다.

Ⅳ. 결론

CBR은 과거의 경험을 이용해서 문제를 해결하는 패러다임으로서 추론자에게 완전하게 이해되지 못한 영역에 대한 해결안의 제공, 해결안 평가 알고리즘이 없을 경우 해결안의 평가, 불명확한 개념의 해석, 반복적인 실수의 예방, 문제의 중요부분에 대한 집중 등을 할 수 있도록 해준다(Miyashita 등, 1996). CBR이 이러한 장점들을 가지고 있다고 할지라도 CBR이 RBR과 MBR에 대해 모든 분야에 있어서 대체수단이 되는 것은 아니다. CBR의 기능이 강화되어 갈수록 문제해결 경험과 도메인 지식이 풍부한 문제영역에 있어서는 CBR이 RBR과 MBR의 효과적 대체수단이 될 수 있을 것이다. 그러나 도메인 지식만이 풍부한 영역에서는 여전히 RBR과 MBR의 역할이 중요시 될 것이다. 따라서 이들 추론방법들은 의사결정자의 문제해결을 지원하기 위해 상호보완적인 역할을 수행할 것이다. 그러나 CBR의 의의는 전문가시스템의 개념을 불완전한 지식을 갖는 영역으로 확장 가능하게 해준다는 것이다.

CBRS는 정적인 측면에서 사례의 표현과 구조화문제가 중요시 되며, 동적인 측면에서는 사례의 검색절차와 수정이라는 해결안 생성과정이 중요시 된다. 그러나 CBR에 대한 연구는 아직 초기단계라 사례에 표현되어야 하는 정보의 내용과 사례들이 어떤 구조로 연결되어야 하는가에 대한 결론이 내려지지 않았다. 본 논문은 이러한 문제점들을 해결하기 위해 정적 측면에서 효과적인 CAPP 지원을 위해 사례의 표현시 고려해야 할 사항들을 고찰한 후 이를 기초로 해서 사례들을 표현하고, 계획수립문제를 탐색의 문제로 간주하여 계산의 복잡성을 줄이기 위해 추상화개념을 활용해서 계층적으로 사례베이스를 구조화하였다. 또한 사례의 구조화시 시스템의 문제해결 능력을 향상시켜주기 위하여 사례베이스를 응용도메인 종속적인 것과 독립적인 것으로 분리하여 과거의 문제해결 경험에 관한 지식은 DDCB에 나타내었으며, 도메인 전문가가 가지는 일반적인 문제해결 지식은 DICB에 나타내었다.

<참고문헌>

1. Aamodt, A., and E. Plaza, "Case Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches," *Artificial Intelligence Communications*, Vol.7, No.1, 1994, pp.39-59.

2. Alterman, R., "Panel Discussion on "Case Representation"," *Proceedings : Case-Based Workshop*, May, 1989, p.15.

3. Alting, L., and H. Zhang, "Computer Aided Process Planning : the state-of-the-art survey," *INT. J. PROD. RES.*, Vol.27, No.4, 1989, pp.553-585.

4. Fedorowicz, J., and G. B. Williams, "Representing Modeling Knowledge in an Intelligent Decision Support Systems," *Decision Support Systems*, Vol.2, 1986, pp.3-14.

5. Gindy, N. N. Z., "A hierarchical structure for form features," *INT. J. PROD. RES.*, Vol.27, No.12, 1989, pp.2089-2103.

6. Gupta, U. G., "How Case-Based Reasoning Solves New Problems," *INTERFACES*, Vol.24, No.6, 1994, pp.110-119.

7. Gupta, N., and Nau, D., "Optimal Block's World Solutions are NP-Hard, *Technical Report*, Computer Science Dept., Univ. of Maryland, 1990. in : Hendler, J., A. Tate, and M. Drummond, "AI Planning : Systems and Techniques," *AI Magazine*, Summer, 1990, pp.61-77.

8. Hendler, J., A. Tate, and M. Drummond, "AI Planning : Systems and Techniques," *AI Magazine*, Summer, 1990, pp.61-77.

9. Irani, S. A., H.-Y. Koo, and S. Raman, "Feature-based operation sequence generation in CAPP," *INT. J. PROD. RES*, Vol.33, No.1, 1995, pp.17-39.

10. Kolodner, J. L., "Judging Which is the "Best" Case for a Case-Based Reasoner," *Proceedings : Case-Based Reasoning Workshop*, May, 1989, pp.77-81.

11. Kolodner, J., *Case-Based Reasoning*, Morgan Kaufmann, 1993.

12. Korf, R. E., "Planning as Search : A Quantitative Approach," *Artificial Intelligence*, Vol.33, 1987, pp65-88.

13. Liang, T., "Analogical reasoning and case-based learning in model management systems," *Decision Support Systems*, Vol.10, 1993, p.137-160.

14. Liang, T., and B. R. Konsynski, "Modeling by analogy - Use of analogical reasoning in model management systems," *Decision Support Systems*, Vol.9, 1993, pp.113-125.

15. Mahapatra, R., and A. Sen, "Case Base Management Systems : Providing Database Support to Case-based Reasoners," *J. of Database Management*, Vol.5, No.2, 1994, pp.19-29.

16. McGovern, J., D. Samson, and A. Wirth, "Using case based reasoning for basis development in intelligent decision systems," *European J. of Operational Research*, Vol.74, 1994, p.40-59.

17. Miyashita, K., K. Sycara, and R. Mizoguchi, "Modeling ill-structured optimization tasks through cases," *Decision Support Systems*, Vol.17, 1996, pp.345-364.

18. Mott, S., "Case-Based Reasoning : Market, Applications, and Fit With Other Technologies," *Expert Systems With Applications*, Vol.6, 1993, pp.97-104.

19. O'Leary, D. E., "Verification and Validation of Case-Based Systems," *Expert Systems With Applications*, Vol.6, 1993, pp.57-66.

20. Paek, Y., J. Seo, and G. Kim, "An expert system with case-based reasoning for database schema design," *Decision Support Systems*, Vol.18, 1996, pp.83-95.

21. Plaza, E., "Cases as terms : A feature term approach to the structured representation of cases," *Lecture Notes in AI*, 1010, Springer Verlag, Heidelberg, 1995, pp.265-276.

22. Rogers, J. S., P. A. Farrington, B. J. Schroer, and R. G. Hubbard, "Automated Process Planning System for Turned Parts," *Integrated Manufacturing Systems*, Vol.5, No.4/5, 1994, pp.41-47.

23. Salomons, O. W., F. J. A. M. van Houten, and H. J. J. Kals, "Review of Research in Feature-Based Design," *Journal of Manufacturing Systems*, Vol.12, No.2, pp.113-132.

24. Slade, S., "Case-Based Reasoning : A Research Paradigm," *AI Magazine*, Spring, 1991, pp.42-52.

25. Vargas, J. E., and S. Raj, "Developing maintainable expert systems using case-based reasoning," *Expert Systems*, Vol.10, No.4, 1993, pp.219-225.