

디자인을 위한 지식기반시스템의 이론적 고찰

A Theoretical Study on the Knowledge-Based System for Design

김태현*/Kim, Tae-hyun

Abstract

Artificial Intelligence is generally concerned with tasks whose execution appears to involve some intelligence if done by humans, and knowledge-based system (in other word, expert system) is the research about the specific domain. This concept also can be applied to interior design field.

So the purpose of this study is in reconstructing the accomplishment of artificial Intelligence and knowledge engineering, searching basic theories and cases to formulate knowledge-based design system, and testing the possibilities how the design information can be dealt in computer system.

Given that recognition, two major problems must be solved before knowledge-based CAD systems could be-

come practical: Firstly, identification of the interior designers use. Secondly, representating this knowledge in a computationally effective manner.

I had discussed the basic concepts on which to base a knowledge-based design model, knowledge representation schemes, and problem solving. I could find the possibility which the knowledge-based system can be applied to the interior design according to this study. But there are non-deductive, often irrational and not easily computerized design process in interior design. Those are problems which are relevant to the machine learning and the creativity in design. So there should be a lot of research about the machine learning and the creativity in design in order to construct successfully intelligent knowledge-based design system.

1. 서론

인간이 가지고 있는 지적인 능력을 기계에 구현하고자 하는 연구를 인공지능이라 하며, 이러한 내용을 어느 전문분야에 적용함으로써 전문적인 지식을 형식화하여 컴퓨터 내에 표현하고 그것을 적절히 사용하는 데 중점을 둔 연구분야를 지식기반시스템 또는 전문가시스템이라 한다. 이러한 지식기반시스템은 실내디자인 분야에도 적용될 수 있다.

따라서 본 연구의 목적은 인공지능과 지식공학에서 이루어진 성과들을 재정립하여, 지식기반디자인시스템을 형성하기 위한 기초적인 이론과 적용사례들을 검토함으로써, 디자인 지식들을 어떻게 컴퓨터 상에서 처리할 수 있는지의 가능성을 모색해 보는데 그 의의가 있다.

이러한 인식에서 볼 때, 지식기반 디자인 시스템이 실용적으로 되기 전에 다음과 같은 두 가지의 중요한 문제들이 선결되어야 한다. 첫째, 디자이너가 사용하는 디자인 지식을 확인하고, 둘째, 유용한 전산적 방법으로 이러한 지식을 표현할 수 있어야 한다.

2. 인공지능과 지식공학

2-1. 인공지능

지능이란 인간이 가지고 있는 지각, 인식, 이해, 기억, 판단, 학습, 사고, 발견, 창조 등과 같은 지적인 일들을 총칭하는 말이다. 이와 같은 능력을 인공적으로 구현하여 기계에 부여하게 되면, 그 기계는 인공지능을 가진 기계가 되며, 지적인 기계라고 부를 수 있다. 이와 같

이 지적인 능력을 기계에 구현하려고 하는 연구가 바로 인공지능 연구이다.

인공지능의 공학적인 접근에서 가장 먼저 직면하게 되는 것은 인간의 사고나 이해를 위한 지식을 어떻게 컴퓨터 상에 구현할 것인가 하는 지식표현 방법과 지식처리 방법이 중요한 과제가 된다. 따라서 인공지능이 종래의 공학과 크게 다른 점이라면, 종래의 공학은 물리, 화학을 기초로 하는데 반해서, 인공지능은 언어학, 철학, 심리학을 배경으로 한 지식정보를 기초로 한다는 점이다.¹⁾

인공지능은 유연한 프로그래밍 양식을 지원하고, 고도의 모듈구조로 된 프로그램을 만들 수 있는 프로그래밍 패러다임을 발전시켜왔다. 이러한 프로그래밍 패러다임들은 프로그램 구조를 통해서 변화의 분기점들을 고려할 필요없이 국부적으로 프로그램을 연장시키고 변형시키는 것을 쉽게 한다. 이러한 프로그램들은 문제를 보다 잘 이해할 수 있도록 도와준다. 그것들은 디자이너의 일상적인 행동 가운데 중요한 역할을 하는 비알고리즘적이고 경험적인 지식을 정확하게 이용하기 위해서 의도된 것들이다. 인공지능은 이러한 유형의 지식과 그 지식으로 추론하기 위한 일반적인 추론 기체들을 표현하기 위해서 계획들을 발전시켜 왔다.

인공지능은 일반적으로 인간의 행위 가운데 지능을 포함하는 작업들에 연관되어 있다. 디자인 행위는 바로 이러한 범주에 들어간다.

인공지능연구는 간략하게 다음과 같이 두 가지의 방향으로 나누어진다.

1) 인간의 마음이 어떻게 작용하는지를 이해하려고 하는 노력. 이러한 노력들은 이론적으로 유도된 경험적인 확증을 요구한다.

*정회원, 빈차디자인 이사

1) 김 현숙, 인공지능이란 무엇인가, 크라운출판사, 1995, p.12

2) 지능적인 작업을 효과적으로 수행할 수 있는 기계를 발견하고자 하는 노력. 이러한 노력들은 실용적으로 유도되어야 하고, 실용적인 유용성 실험을 거쳐야 한다.

이렇게 서로 다른 유도들로부터 생성된 결과들은 대조적인 방법에 의해서 평가되어야 한다.

생성시스템은 다른 사람의 작업으로부터 이점들을 얻기 위한 것으로서, 인간의 기억을 모형화하려는 관심에 의해서 유도된 것이다.

전통적인 개념의 CAD 시스템은 어떤 영역의 전문적인 지식을 요구하는 작업만을 취급하는 인공지능과 디자인 분야의 영역 밖에 존재한다. 어떤 형식으로든 디자인 전문분야에 접근하기 위한 프로그램은 다음과 같은 두 가지의 기본적인 방법들로 인간 디자이너를 보완하고 있다. 첫째로 그들이 고려해 보지 못한 가능성들을 제시하고, 둘째로 그들이 잊어버렸던 것들을 상기시킨다.²⁾

2-2. 지식공학

지식공학은 정보공학의 한 분야인 인공지능의 응용분야로서, 인간이 사고하듯 컴퓨터도 사고할 수 있게 하기 위한 발상에서 컴퓨터에 지식을 내장하고 인간의 지적인 활동을 지원할 수 있는 시스템을 만들어 내는 기술을 제공하는 공학이다. 즉, 인간이 가지고 있는 일상적인 지식 및 상식 그리고 대상세계에 대한 전문적인 지식 등을 어떻게 형식화하여 컴퓨터 내에 표현하고 동시에, 어떻게 적절히 사용할 수 있는가에 중점을 둔 지식표현과 그 이용을 목적으로 한다.

이전의 인공지능 연구는 게임이나 퍼즐 등과 같이 닫힌세계를 대상으로 하고 추론 중심형의 연구가 이루어지고 있는데 반해, 지식공학은 교과서적인 사실에 관한 지식(facts)과 전문가의 오랜 세월을 걸친 학습과 연구 및 경험 등을 통해서 획득한 경험적 지식을 이용하여 전문분야용 전문가 시스템을 구축하고, 고도의 문제해결(Problem Solving)에 적용할 경우 높은 수준의 성능을 기대할 수 있는 컴퓨터 응용공학이다.

지식공학과 인공지능, 인지과학 등의 각각의 연구영역은 상호간에 그다지 명확하게 구별할 수 있는 성질의 것이 아니나, 대체적으로 다음과 같은 양상을 띤다. 지식공학은 지식을 가진 컴퓨터를 어떻게 만들어낼 것인가에 관심이 있으며, 전문가시스템의 연구자들이 비교적 많다. 인공지능에서는 지식의 표현이나 내용만이 아니라 지적 시스템에 대해 보다 폭넓은 입장에서의 검토가 이루어지고, 패턴인식이나 자연어처리 등의 연구자들이 더 많다. 인지과학에서는 인간의 지식이나 이해가 도대체 어떠한 구조로 되어 있는가에 관해 연구가 진행되고 있으며, 심리학자나 언어학자, 철학자 등이 보다 많다.

전문가 시스템은 문제해결의 연구에 방향성을 제공하고 동시에 그 실용성을 테스트하기 위한 하나의 매체로서 사용된다. 따라서 전문적인 지식이 없으면 풀 수 없을 만큼 어려운 실제적인 문제들을 풀기 위한 문제해결 프로그램을 의미하며, 이와같은 시스템은 그 성능이 전문가가 사용하는 사실이나 경험에 크게 의존하기 때문에 지식기반시스템이라고도 부른다.

3. 지식기반 디자인 시스템

3-1. 지식기반 디자인의 모형

디자인 프로세스의 지식기반 모형의 목적은 컴퓨터로 하여금 디자인 프로세스를 지원할 수 있도록 하기 위한 것이다. 지식 기반 모형의 기초를 이루는 세 개의 개념이 있다.

1) 표현: 컴퓨터 시스템에서 정보를 표현하는 방법.

2) 추론: 디자인 추론에 의해 이해되는 것.

3) 구문: 디자인과 컴퓨터 시스템에서의 구문지식의 역할.

디자인 모형의 분리 가능한 두 개의 작업은 해석과 생성이다. 해석이란 디자인 묘사와 그것의 성능 요구 사이의 사상이며, 생성이란 디자인의 구성을 말한다. 이렇게 간단한 개념으로부터 디자인의 다양한 과정들을 설명할 수 있고, 디자인 시스템을 위해서는 어떤 기능들이 필요한지를 구체화시킬 수 있으며, 그러한 디자인 시스템을 구성하기 위한 방법을 깨닫게 한다.³⁾

3-1-1. 표현

정보표현에 이용되는 여러 가지의 방법들의 기반을 이루고 있는 개념들을 다루기 위해서는 사실, 지식, 그리고 제어 등의 개념을 정의할 필요가 있다. 이러한 정의를 표현하는 것은 정보처리의 논리기반 관점을 묘사하는 것과 같다. 여기서의 주요한 관심은 지식을 사실들 사이의 관계에 대한 진술로 여기는 형식적인 관점에 있으며, 이러한 개념은 디자인 추론에서 지식의 역할을 고려하게 한다.

1) 사실 (Facts)

디자인에 속하는 정보가 컴퓨터 시스템에서 조직화 될 수 있도록, 정보를 두 가지의 유형(객체와 관계)으로 분할하는 조직체제를 고려해보자. 객체(Object)는 간단히 정보의 단위이다. 이러한 정의를 디자인에서 흔히 사용하는 용어로서의 오브젝트와 구별할 필요가 있다. 디자인에서의 오브젝트는 기둥, 가구, 부엌 등과 같이 물리적이거나 공간적으로 분리된 요소를 말한다. 관계는 정보 단위들의 그룹에 붙여진 라벨로써, 물리적인 객체들 사이의 관계성으로 설명해도 좋다. 물리적 관계의 예로써 "~옆에", "~의 안에", "~보다 긴" 등을 들 수 있다. 객체들의 그룹을 묘사하기 위해 관계라는 용어를 사용하는 것은 디자이너의 세계에서는 자유로운 것이다.

이러한 유형의 정보조직을 기호화하여 객체와 관계로 표현하는데는 여러 방법들이 존재한다. 하나의 유용한 방법은 하나의 그룹에 속한 객체들의 이름을 괄호 안에 넣고, 관계를 괄호 앞에 놓는 것이다.

내부 (책상, 방)

옆 (거실, 식당)

보다 큰 (안방, 건너방)

위의 사실은 아래와 같이 자연어로 해석될 수 있다.

책상이 방 내부에 있다.

거실이 식당 옆에 있다.

안방은 건너방 보다 크다.

사실은 객체와 관계의 그룹이며, 또한 다른 정보의 단위와 그것의 관계에 대한 것일 수 있다. 또한 값을 지닌 사실 - 그것이 참이든 거짓이든지 간에 - 에 대해서도 표현할 수 있다.

층고 (거실, 2300)

색상 (벽체1, 회색)

이것들은 다음과 같이 해석된다.

거실의 층고는 2300이다.

벽체1의 색상은 회색이다.

같은 정보가 여러 가지 방법으로 기록될 수 있다. 예를 들면, 어느 특정한 의자(의자1)는 가구나 불리는 객체들의 클래스에 속한다라고 사실을 표현하는 것은 편리할 것이다. 이것은 최소한 두가지의 방법으로 표현될 수 있다.

클래스 (의자1, 가구)

가구 (의자1)

2) Flemming, U. "Artificial Intelligence and Design : a Mid-Term Review" in Knowledge-Based Computer-Aided Architectural Design, Elsevier, 1994, p.4.

3) Coyne, R. D., M. A. Rosenman, A. D. Radford, M. Balachandran, and J. S. Gero. Knowledge-Based Design Systems, Addison-Wesley, New York, 1990, p.43.

마지막 예에서 클래스의 이름은 관계로써 표현되었다. 이와 같은 표현방법은 속성에도 적용될 수 있다.

속성 (실1, 정사각형 평면)

정사각형 평면 (실1)

정보를 객체와 관계로 나누는 것은 간단한 개념이지만, 이러한 유형의 조직을 순환적으로 적용하면 풍부하게 전개될 수 있다. 객체는 정보의 단위를 표현하지만, 또한 사실이 될 수도 있다. 어떠한 형태의 표현이 선택되더라도, 사실은 정보의 단위들과 그 단위들 사이의 관계에 의해서 묘사될 수 있다.

2) 지식

객체와 관계들은 모든 정보구조가 정의될 수 있는 정보처리시스템의 기초적 요소가 된다. 비록 객체와 관계들은 매우 간단한 개념이지만, 우리가 보통 지식으로 언급하는 것을 이해하는데 기본적인 역할을 한다. 지식은 사실들 사이의 사상에 대한 진술들이라는 것으로 특징지을 수 있다. 이러한 사상은 기존의 사실들로부터 새로운 사실들을 추출할 수 있도록 도와준다.

이러한 사상을 표현하는데는 많은 방법들이 있다. 가장 유용한 방법 중의 하나는 사실들 간의 합리적인 연결을 나타내는 일련의 구체적인 관계들을 정의하는 것이다. 그러한 논리적 관계들의 예로는 "그리고(and)", "또는(or)", "만일(if)" 등이 있다.

정보체계에서 기호화되는 것을 원하는 지식의 항목은 객체가 다른 객체에 의해서 지지되는지에 의해서 결정된다. 예를 들어서 어떤 객체가 다른 객체 위에 있다면, 그 객체는 다른 객체에 의해서 지지된다고 말할 수 있다. 바닥위에 카펫이 깔려져 있고, 그 위에 테이블이 있으며, 그 테이블 위에 컴퓨터가 있다고 가정해보자. 이때 컴퓨터는 바닥에 의해 지지된다고 볼 수 있다. 왜냐하면 컴퓨터는 바닥에 의해서 지지되는 객체들 위에 있기 때문이다. 이러한 지식을 객체와 관계에 의해서 표현될 수 있는 방법중의 하나는 다음과 같이 "만일(IF)"과 "그리고(AND)"라는 관계들을 포함하는 괄호 속의 사실들에 의해서 이루어질 수 있다.

If (SupportedBy(a,b),On(a,b))

If (SupportedBy(a,b),And(On(a,c),SupportedBy(c,b))

이것을 좀더 자연스럽게 만들면 다음과 같다.

SupportedBy(a,b) If On(a,b)

SupportedBy(a,b) If On(a,c) And SupportedBy(c,b)

사실들 사이의 사상인 지식에 의해 의미하고자 하는 것은 일반화로서의 지식개념과 일치한다. 일반화는 객체들의 클래스에 관한 진술이다. 즉 일반화는 단순히 클래스 구성원에 관한 사실들 사이의 관계에 대한 진술로서 정보저장소에서 무가치적인 경제성을 표현한다. 구적진술(clausal statement)에 의한 일반화의 예는 다음과 같다.

Furniture(x) If chair(x)

이것은 만일 어떤 객체 x가 의자라고 불리는 클래스에 속한다면, 객체 x는 가구라고 불리는 클래스에 속한다는 것을 말한다.

지금까지 지식은 논리적인 구들(logical clauses)로써 논의되었다. 구형식(clausal form)의 이점은 지식의 개념을 사실들 사이의 관계로 명확하게 하는 것이다. 지식처리하는 지식이 속해 있는 영역과는 독립적인 문제로 여겨질 수 있으며, 지식을 전산처리에 의해서 고려해 볼 수 있다. 구형식은 컴퓨터 프로그램의 제어 구성과는 별도로 지식의 기본적인 특성을 조명해 보는데 도움을 준다.

3) 제어

제어란 지식항목들 사이의 사상에 의해서 특징지어진다. 논리적 과정을 논리 자체의 수단에 의해 묘사하는 것은 가능하다. 그러므로 논리는 그 자신의 작용이나, 또한 다른 논리들의 작용을 묘사하기 위한 메타 언어로서 제공된다. 그 이점은 컴퓨터 시스템에서 지식의 사용을 제어하는 방법을 메타지식의 문제로 고려해 볼 수 있다는데 있다. 메타지식은 단순히 지식 진술들 사이의 관계 - 즉, 구들 사이의 관계 -

와 연관되어 있다.

제어지식은 구들 사이의 관련성을 다루는 특정한 관계(메타 논리적 관계)를 한정함으로써 묘사될 수 있다. 논리적 구들을 처리하기 위한 메타지식체계는 특정한 구들에 의하여 표현될 수 있다. 만일 우리가 엄격한 논리로부터 벗어나 논리적 프로그래밍으로 알려진 컴퓨터 실행을 고려한다면, 이러한 유형의 표현 값은 매우 명확하다. 논리구들의 일반적인 선언적 해석과 같이, 논리 프로그래밍은 그것의 절차적 해석에 연관되어 있다. 이것은 단순히 "구의 체계(body of a clause)"에서 사실들의 순서가 중요하다는 것을 의미한다. 지식이 절차적으로 다루어지는 방법을 감독하기 위해 메타 논리가 이용된다. 그러한 개념은 메타제어의 고려 즉, 제어진술들 사이에서의 관련성에 속하는 지식에까지 연장될 수 있다.

지금까지 객체들과 관계들에 의한 정보가 어떻게 컴퓨터 내에서 표현될 수 있는 지에 관한 개념적 스키마를 증명하고자 시도하였다. 이러한 두 개념들은 우리가 정보를 표현하고자 하는 방법 - 사실들 간의 관계성에 대한 진술(소위 지식)로서, 그리고 지식(메타지식이나 제어지식)에 대한 진술로서 -에 풍부하게 사용되고 있다.⁴⁾

3-1-2. 추론

디자인 행위가 추론을 일반적으로 이해하는 방법과 어떻게 일치하는지를 보기 위해 논리학적 방법인 연역법과 귀납법, 그리고 유도법으로 분류하여 정리하면 다음과 같다.

1) 연역법

논리학상의 연역은 우리가 확신을 가지고 논의할 수 있는 추론방식이다. 이런 유형의 추론은 증명에 적합하고 쉽게 형식화될 수 있다. 연역은 구적 진술을 처리하기 위한 수단으로 제공된다. 연역의 작용은 3단 논법에 대해 최초로 기록을 남긴 아리스토텔레스에 의해서 입증되었다. 3단 논법은 추론의 기본적인 특성에 대해 언급할 때 편리함을 제공한다. 간단한 3단 논법을 다음과 같이 설명할 수 있다.

a) 이것은 의자이다.

b) 모든 의자는 가구이다.

c) 그러므로 이것은 가구이다.

3단 논법의 구성원들은 다양하게 격, 규칙, 결과, 또는 전제, 공리, 정리 등으로 분류되며, 정보 스키마에 의해서는 사실, 지식, 추론된 사실 등으로 분류된다.

2) 귀납법과 유도법

귀납법은 절대적으로 필요한 인간의 추론 행위로서, 과학에서의 귀납법은 어떤 현상의 관찰로부터 이론이 유도되는 과정으로 여겨진다. 과학의 영역 안에서 3단 논법의 구성원들은 관찰 진술, 이론, 그리고 추측 등으로 이루어질 수 있다. 이러한 관점에 따르면, 이론은 관찰 가능한 현상의 어떤 클래스들을 묘사하기에 충분한 일반화나 지식의 체계로 설명된다.

유도법은 원칙적으로 피어스(Peirce)에 의해서 제안된 것으로서, 논리적인 규칙과 결과들이 주어진 세계에 대한 진술들을 유도하는 것이다. 3단 논법에서 만일 우리가 "모든 의자는 가구이다."라는 규칙과 "이것은 가구이다."라는 결론을 알았고, "이것은 의자이다."라는 초기의 진술을 몰랐더라도 유도법은 작용할 것이다. 우리가 결정해야 할 단계인 "이것은 의자이다."는 유도적이다. 이러한 추론 단계는 다른 유형의 가구가 될 수도 있다. 연역적인 시스템을 후진으로 실행하는 것은 일반적으로 정확하지 않다.

3) 해석과 디자인 생성

지금까지 어떻게 디자인이 사실들에 의해서 컴퓨터 데이터베이스로 묘사될 수 있는 지에 대해서 논의해 왔다. 그러한 묘사는 인공물의 물

4) Coyne, R. D., M. A. Rosenman, A. D. Radford, M. Balachandran, and J. S. Gero. 앞의 책 pp.50-51.

리적인 구성원들에 대한 진술들과 구성원들 사이의 관계성으로 되어 있을 것이다. 이러한 것은 전통적인 CAD시스템에 의해 생성되는 것과 같은 전형적인 데이터베이스를 구성하고 있다. 우리는 연역법을 그러한 데이터베이스의 해석과 관련지음으로서 특성화시킬 수 있다. 이것은 CAD 시스템의 일반적인 관심이다. 해석지식은 논리적인 규칙이나 다른 표현 구조의 형식으로 명확하게 만들어 질 수 있으며, 이러한 지식은 절차적 컴퓨터 프로그램 내에서 감춰질 수도 있다.

그러므로 인공물 묘사를 해석하기 위한 지식을 표현하는 것은 가능하다. 그러나 디자이너는 우선적으로 묘사를 생성하는데에 관심을 가지고 있다. 우리는 디자인할 때 일반적으로 인공물 묘사로 시작하지 않고, 성능에 대한 개념들로 시작한다. 그리고 나서 우리는 기하학과 재료의 속성들로 묘사하기 위해 노력한다. 이것은 매우 유도적인 과정과 유사하다.

3-1-3. 구문

지금까지 사실들 사이의 사상과 관련하여 지식-새로운 사실들이 기존의 것들로부터 추론될 수 있는-을 논의해 왔다. 만일 사실 A가 진실이면, B도 또한 사실이라고 어떤 규칙이 설명될 것이다. 이러한 기반 위에 형성된 해석시스템에서 사실들의 개수는 새로운 추론들이 만들어질 때마다 증가한다. 그러나 사실들 간에는 다른 유형의 사상이 존재한다. 만일 디자인에 관한 사실들이 참이면, 그것들은 다른 사실들에 의해 대체된다고 진술하면서 규칙이 형성될 수 있다. 왜냐하면 이러한 유형의 규칙은 일련의 사실들-즉, 디자인 묘사-의 상태를 변화시키며, 이것은 또한 행동이라고 불린다. 행동을 바라보는 또 다른 방법의 시각은 고쳐 쓰는 규칙(rewriterule)이다. 전산이론에 따르면, 고쳐 쓰는 규칙은 언어의 구문을 한정하는 문법체계의 기반을 형성한다.

3-2. 디자인 지식표현안 사례 검토

이 부분에서는 디자인 적용에도 유용한 것으로 입증된 인공분야에서 발전된 지식표현 계획안들을 검토할 것이다. 각각의 안들은 지식을 표현하기 위해 작은 모듈이나 큰 덩어리(chunk)-지식을 표현하기 위한 프로그래밍 언어-의 형태로 구체적인 형식적 구성을 제공한다. 표현의 모듈성은 개발자들이 지식기반의 내용에 집중하고 점증적으로 그것을 구축할 수 있도록 해준다. 반면에 지식과 제어 사이의 분리는 프로그램이 즉시 실행되는 것을 보장하고, 처음부터 지식획득의 과정을 지원하는데 이용될 수 있다.⁵⁾

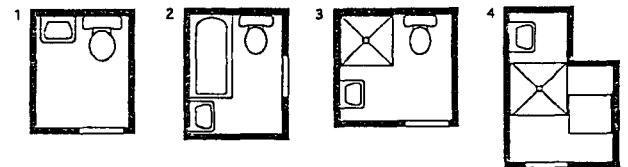
1) 규칙과 생성시스템

규칙은 어떤 행동이 일어나는 조건이나 구문을 설명하는 왼쪽 부분(LHS) 혹은 IF 부분과, 행동을 묘사하는 오른쪽 부분(RSH) 혹은 THEN 부분으로 구성되어 있다.⁶⁾ 디자인 구문에 있어서 규칙은 고도의 경험을 지닌 전문가의 특정한 격추론 특성을 전형적으로 포착하여 작용적으로 만든다.⁷⁾ 규칙의 명세를 지닌 시스템은 규칙의 적용으로 변화될 수 있는 활동적인 메모리를-때때로 데이터 베이스로 불리는데- 포함하고 있다. 규칙의 LHS가 활동메모리의 어느 부분과 일치하면 그 규칙은 적용될 수 있다. 적용 자체는 RHS에 의해 결정되고, 활동 메모리의 어떤 부분을 변화시킨다. 규칙을 적용시킬 수 있도록, 그러한 시스템은 추가적으로 규칙 해석기-적용 가능한 규칙을 찾을 수 있고, 그것들을 실행시킬 수 있는-를 갖추어야 한다. 하나 이상의

규칙이 적용 가능할 때는 해석기가 선택할 수 있어야 한다. 이런 경우에 규칙해석은 충돌도출(Conflict Resolution)이라 불린다.

규칙들의 집합과 활동 메모리, 그리고 규칙 해석기 등으로 구성되는 시스템을 생성시스템이라 부른다.⁸⁾ 생성시스템은 여러 가지의 이유로 디자인 구문에서 매력적으로 보인다. 생성시스템은 고도의 모듈러이다. 규칙들은 간접적인 활동 메모리의 내용을 통해서만 의사소통할 수 있기 때문에 개별적으로 정의되고 변경될 수 있다. 그러므로 생성시스템은 작동할 때는 완전히 독립적이다. 생성시스템은 전형적으로 구별될 수 있고 조사될 수 있는 한 개의 지식 덩어리를 포착한다. 이러한 모듈성은 지식 기반을 점증적으로 구축한다. 전문가가 그 시스템의 작동을 관찰할 때, 규칙들에 삽입되어야 할 지식이 발견되고 구체화되며, 결과적으로 규칙으로 표현될 수 있다. 현재 상태의 활동 메모리에 근거하여 동적으로 규칙들을 야기하는 규칙 해석기에 의해서 또다른 이점이 형성된다. 즉, 규칙 해석기는 그 과정 자체에 반응하면서 유연하게 규칙 적용의 결과들을 생성한다. 그 결과들은 절차 프로그램에서 요구되는 것처럼, 프로그래머에 의해서 예상될 필요가 없다.

이러한 이점들은 <그림 1>의 화장실 가구배치 디자인을 위한 생성 전문가 시스템인 LOOS 시스템에서 잘 나타나 있다.⁹⁾ 이 시스템의 주요한 작업은 두 구성물-생성기와 실험기-에 의해서 이루어진다. 생성기는 한번에 하나의 물체를 첨가함으로써 배치를 점진적으로 구축하고, 실험기는 사용자가 구체화시킨 제어들이나 기준들에 따라서 배치를 평가한다. 이 시스템의 첫 번째 버전은 생성시스템 언어인 OPS83 으로 프로그램되었다. 실험기는 어느 특정 응용 영역의 관심사를 반영하며, 그 시스템과 함께 작업하고 그것의 작동을 관찰함으로써 발견되어지는 실험규칙들의 집합을 구성하고 있다.



<그림 1> 플래닝과 그의 동료들에 의한 LOOS 시스템의 적용을 위해 실험기를 구축하는데 이용된 사례

2) 논리적 프로그래밍

논리적 프로그래밍은 프로그래머가 논리적인 서술어들의 형식으로 된 사실들과 규칙들을 통해서 어떤 영역에서 참인 것으로 여겨지는 것을 표현할 수 있도록 해준다. 사실들과 규칙들을 구성하고 있는 논리적 프로그래밍은 무엇이 참인가, 또는 무엇이 암시될 수 있는가 등의 질문에 대답할 수 있다. 이러한 질문들은 두 개의 내장된 장치들에 의존하고 있다. 그것들은 도출과 통합이다. 도출은 사실과 서술어의 짝짓기를 통해서 하나의 술어를 설정하기 위한 시도이다. 이것은 서술어 각각의 순환적인 도출이나 백트래킹을 요구할 것이다. 통합은 서술어들을 그와 동등한 서술어들로 대체하고, 도출하는 동안에 값들에게 변수들을 연결시킨다. 예를 들어 만일 사실들의 집합이 선분들(선들의 끝점의 좌표들을 통한)의 집합을 묘사하고, 네 개의 선분들이 하나의 사각형을 형성한다는 조건을 명기하는 규칙이 있다면, 그 프로그램에

5)Flemming, U. 앞의 책, p.5.

6)생성시스템은 A. Newell팀이 인간의 인지모델에 기초한 문제해결의 한 방법으로써 고안한 것이다. 지식은 [IF 조건 THEN 행동(실행)]이라는 형식으로 간단하게 표현된다.

7)Hayes-Roth, F. "Rule-Based System," Comm. of the ACM, 1985, pp.921-932.

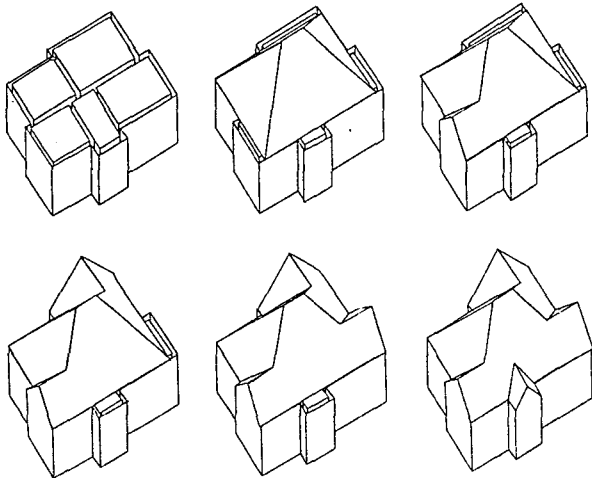
8)Davis, R., and J. King. "An Overview of Production Systems," in Machine Intelligence 8, Halstead Press, New York, 1977, pp. 300-332.

9)Flemming, U., R. Coyne, T. Glavin, and M. Rychener. "A Generative Expert System for the Design of Building Layout-Version 2," in Artificial Intelligence in Engineering: Design, J. S. Gero (ed.), Elsevier, New York.

대한 반복적인 질문들이 시각형을 형성하는 선분들의 모든 조합들을 검색할 것이고 각각의 경우에 선분의 모서리 좌표에 구체적인 값들을 적용할 것이다.

디자인 구문에서, 그래픽적인 기본 형태들을 통해서 디자인이 구체화되는 방법을 표현할 수 있는 논리적 프로그램의 규칙들은 보다 높은 단계의 영역의 객체들과 개념들의 연속적인 유도에 의해서 해석될 수 있다. 따라서 만일 디자인이나 구문이 주어진다면, 디자인을 해석하거나 디자인들을 유도하기 위해 규칙들이 이용될 수 있다.

논리적 프로그래밍은 피츠버그의 전통보존지역에서의 주거디자인에 관한 플레밍의 프로젝트에 서 잘 나타나 있다. 이 프로젝트는 실질적인 주거계획에 의해 동기 유발된 것으로서, 어떻게 새로운 구조물이 그 지역에 어울릴 수 있는지에 답변하고자 한 것이다. 그는 분석의 도구로서 컴퓨터를 사용하여 그 지역에 만연하는 역사적 패턴들을 이해하고자 하였다. 즉, 그는 그러한 패턴들의 기초를 이루는 원리들에 대한 지식을 컴퓨터 프로그램에 삽입시킴으로써 패턴들에 대한 이해를 실험하고자 하였으며, 따라서 그러한 원리들에 의해서 패턴들을 생성시킬 수 있었다. (그림 2)는 전형적인 빅토리아식 주택의 불규칙한 평면 유형을 위한 기하학적 지붕의 단계적인 생성을 보여주고 있다.¹⁰⁾



(그림 2) 플레밍에 의한 Prolog 규칙을 통한 지붕의 단계적 생성

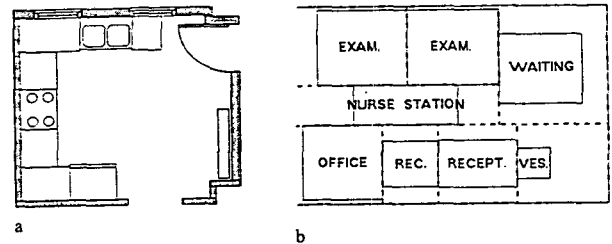
3) 프레임 (Frames)과 객체지향 프로그래밍 (Object-Oriented Programming)

프레임은 객체의 특성을 속성-값의 짝으로 기록하는 표현 안이다. 이러한 안은 물리적이고 형식적인 특성들에 의해 객체들이 특징지어지는 디자인 분야에서는 자연스러운 것이다.

상속개념을 함유한 프레임 표현과 객체지향프로그래밍에서 알려진 폴리모피즘을 결합하는 것은 디자인을 위한 좋은 표현 안을 만들어 낸다. 상속은 클래스의 정의나 분류학적 계층을 가능하게 한다. 즉, 클래스 구성원들과 부클래스들은 그것들이 속한 클래스나 슈퍼 클래스의 정의로부터 속성을 상속받을 수 있다. 폴리모피즘은 전산적인 방법들이 클래스 구성에 의존하여 구체화될 수 있도록 하는데, 어떤 방법을 실행하는데 필요한 지식은 그 클래스안에 있다. 이러한 장치들이 함께 이루어져서 고도의 모듈러 시스템을 내놓는다.

디자인에서 객체지향적 접근의 유리한 점은 플레밍과 그의 동료들의 작업에서 잘 나타나고 있다. 그들은 상용리스프(CommonLisp)의 객체지향적 확장으로 루스시스템(LOOS system)의 두 번째 버전을

만들었다. 첫 번째 버전에서의 최초의 응용 영역은 고정된 치수로 이루어진 객체(가구)들을 포함한 부속공간을 다루었다. (그림 3a) 새로운 개작에서 방들이나 좀더 큰 배치 구성원들과 같은 변할 수 있는 치수들로 이루어진 객체들을 다룰 수 있는 능력을 추가해야 했다. (그림 3b)¹¹⁾ 이러한 개작은 주어진 객체지향 프로그래밍 환경에서는 쉬웠다. 그들은 "디자인 객체"라는 클래스의 부클래스로써 "변할 수 있는 크기의 객체"라는 새로운 클래스를 정의해야 했고, 객체가 삽입된 이후에 유효한 치수의 범위를 결정하는 방법의 구체화된 버전을 개발해야 했다. 고정된 크기의 객체들의 정의와 그것들과 연관된 치수들 재는 방법들은 변하지 않은 채로 남아 있을 수 있었다. 그 시스템은 객체가 속해 있는 클래스에 의존하여 자동적으로 올바른 방법을 선택했다. 즉, 그들은 적절한 선택들을 명확하게 기호화할 필요가 없었다.



(그림 3) 플레밍에 의한 고정적인 치수(a)와 변할 수 있는 치수(b)들로 이루어진 객체들의 배치

4. 문제해결

4-1. 디자인방법론

1960년대 초에 디자인 결과를 개선시키기 위하여 - 디자이너들로 하여금 어떤 절차나 방법론을 따르게 함으로써 - 중요한 시도들이 이루어졌다. 이러한 방법론들은 디자이너들에게 디자인 결과를 제공하는 여러 기술이나 수단들을 포함하고 있다. 또한 그것들은 디자인의 인지를 형성하는데 영향력을 미쳐왔다. 디자인에 대한 작용연구 접근방법은 1940년대와 50년대에 발전하여, 1962년 런던의 임페리얼 대학에서 개최된 디자인 방법론에 관한 회합에서 결집되었으며, 연속적으로 1965년에 버밍햄 심포지움이 개최되고 1968년에 캠브리지에서 디자인 방법론 그룹 회의 등이 개최되었다. 그 이후로 디자인에 대한 다양한 방법론적 접근방법들 - 집합이론(Alexander), 의사결정(Archer), 시스템 분석과 위계적 분해(Mesarovic, Macko, 그리고 Takahara) - 이 발전되어 왔다. 모의실험(Forwood)과 최적화(Radford와 Gero) 등을 포함한 많은 기술들이 디자인 과정에 적용되어 왔다. 그중 많은 기술들이 수학적인 절차를 채택하고 있다.

수많은 디자인 과정 모형들 - Asimow, Luckman, Markus 등에 의해 발전된 모형들을 포함하여 - 이 이렇듯 작업들로부터 발생되어 왔다. 거기에는 세 단계의 디자인 모형에 대해 광범위한 동의가 있어 왔다. 그 단계들은 아시모우에 의해서 형성된 분석, 종합, 평가로 묘사될 수 있다. 분석단계는 진단하고, 한정하며, 준비하는 것이다. 즉 문제를 이해하고 목표에 대해 명확한 진술을 생성하는 것이다. 종합단계는 그럴듯한 해결 찾기를 포함하고 있고, 평가단계는 목표와 관련하여 해결의 타당성을 판단하고 대안들 중에서 선택하는 것을 포함하고 있다. 분석을 다시 검토함으로써 해결이 개정되고 개선되는 것은 순환적이다. 이러한 세 단계들은 디자인 프로젝트를 계획하고 조직하며 전개하기 위한 기초적 뼈대를 형성한다.

10)Flemming, U. "The Role of Shape Grammars in the Creation and Analysis of Design," in Computability of Design, Y. Kalay (ed.), Wiley, New York, 1987, pp. 245-272.

11)Flemming, U. 앞의 책, p.9.

존스(Jones)는 세 국면을 약간 다르게 분기, 변형, 집중 등으로 묘사하고 있다. 분기는 문제를 조각들로 분해하고 공간-해결을 위해 충분한 탐색이 일어나는 -의 한계를 정의하는 것과 연관되어 있다. 변형은 새로운 방법으로 조각들을 함께 집어 넣는 것을 포함한다. 집중은 새로운 배열을 실제로 적용시켜 결과를 발견하기 위해 실험하는 것이다.

또한 이러한 모형은 목표 분해 - 알렉산더의 초기 작업에 의해 공헌된 개념 -에 의해서 논의되어 왔다. 이러한 관점에서 본다면 일반적인 목표는 위계적으로 조절될 수 있다. 최고 단계의 목표는 막연할런지 모르며, 일련의 조작가능한 부목표들로 전환되어야 한다. 문제의 분해는 원초적인 단계로 까지 수행되어야만 한다고 알렉산더는 제안했다. 게다가 그는 다양한 부분적 해결들을 종합하는데 도움을 주는 군집 분석 방법론을 제안했다.

평가는 종합의 뒤에 오고, 분석은 종합에 선행하는 것이기 때문에 세 단계의 과정은 연속적이다. 그러한 연속이 여러번 반복되기 때문에 세 단계의 과정은 순환적이다. 평가는 문제의 수정된 분석을 낳고, 분석은 새로운 일시적인 해결의 종합을 유도해낸다. 각각의 순환은 이전의 것보다 점진적으로 덜 일반적이고 보다 더 상세해진다. 영국의 왕립 건축가 협회는 각 프로젝트를 위한 작업계획을 출간했다. 거기에는 브리핑, 스케치 디자인, 디테일 디자인, 서류 등의 단계들이 명확하게 분리되어 있다. 브리핑은 분석단계이다. 스케치는 제안되고, 평가되고, 개정되며, 좀더 디테일하게 작업에 들어가는 단계이다. 스케치 디자인의 형식화는 또한 어느 정도의 디테일한 분석과, 다양한 시험적인 제안들, 그리고 이러한 제안들의 평가 등을 포함하고 있다.

이러한 모형에 따르면, 종합이 시작되기 전에 디자인 문제가 완전히 분석될 필요가 없다. 완전한 분석이란 매우 간단한 작업을 제외하고는 불가능하다. 만일 우리가 충분한 선정지명을 가지고 있다면, 즉 우리가 디자인 작업의 처음부터 필요한 모든 정보를 가지고 있다면, 문제 분해는 가능할 것이다. 실제로 있어서는 이것은 매우 발생하기가 어렵고 우리가 해결을 위한 탐색을 시작할 수 있도록 충분한 분석을 실행함으로써 진행된다. 이러한 진행을 통해서 차례로 다양한 부분체들이 발생하고 같은 종류의 과정들이 반복된다.

4-2. 디자인 문제해결

비록 과학이 디자인의 모형을 제공한다는 관점은 우리가 될지라도, 포퍼(Popper)의 문제해결 모형으로부터 배울 수 있다. 탐색은 잠재적인 해결들과 부분적인 해결들의 공간을 탐구하는데 연관되어 있고, 가설들을 생성하고 실험하는 것은 탐색의 과정으로 특성 지워질 수 있다. 사이몬(Simon)은 이 과정을 미로를 탐구하는 것과 유사한 것으로 묘사한다. 사람은 마음속에 어떤 목적지 - 즉, 목표 - 를 가지고 있다. 통로들 사이의 교차점들은 해결 상태이며, 목표에 도달할 때까지 합리적인 방법으로 통로를 협상하는 것이 필요하다.

디자인은 상태공간을 통한 탐색의 문제해결과정으로 여겨진다. 그 상태란 디자인 해결을 표현한다. 이러한 상태공간은 많은 상태들이 존재하기 때문에 크고, 상태들 사이에 많은 관계들이 존재하기 때문에 복잡하다. 탐색은 어떤 해결들을 실행 불가능하게 하는 목표들과 제어들에 기반을 둔 의사결정을 포함할 것이다. 크고 복잡한 문제들을 위해서는 계층적 접근방식이 유효하다. 연산연구에서의 어떤 기술들은 숫자적인 영역 내에서의 탐색과 연관되어 있으나, 탐색기술 또한 다른 형태의 상징적 처리에 적용된다.

자동화된 문제해결시스템을 위한 많은 이론적인 자극은 뉴웰과 사이몬의 작품들로부터 기인한다. 그들은 디자인 연구에 많은 영향을 미쳤다. 그들은 범용문제해결기구(GPS)라고 불리는 자동화된 문제해결시스템을 묘사하고 있다. 그와 같은 시스템은 현재의 상황과 요구되는 상황(목표들), 그 둘 사이의 차이점들, 그리고 그 시스템을 좀더 목표

에 가깝게 가져오기 위해서 객체들이나 상황들을 변화시키는 행위들을 표현할 수 있어야 한다고 그들은 요구조건들을 진술하고 있다. 그 시스템은 또한 차이점들을 줄이거나 제거할 것 같은 행위들을 선택할 수 있어야 한다. 범용문제해결기구는 과정을 평가하고 대안적인 패스들을 실험하는 수단들을 제공하는 차이점들과 행위들 사이의 연결표를 구체화한다. 수단 - 목적 장치(means-ends mechanism)는 탐색을 보다 효율적으로 하게 한다. 그와 같은 몇몇의 제어장치는 탐색의 양을 제한하고, 시스템의 활동을 옳은 방향으로 향하게 하기 위하여 일반적으로 필요하다. 디자이너들이 가능성들을 탐구하는 방향을 고려해 볼 때, 디자인은 일반적으로 약간의 탐색과정을 포함한다는 것은 명백하다. 적절한 탐색전략들을 유도하는 것은 인공지능연구에서 상당한 주목을 받고 있다. 아킨(Akin)은 "언덕 오르기"와 경험적 탐색, 그리고 전진추론 등과 같은 인공지능에서 사용되는 기술들과 유사한 디자이너들의 행동에서 탐색전략을 관찰했다.

디자이너가 본질적으로 잘못 구성되어 있을 때, 디자이너는 알맞게 조직된 부작업들로 작업을 줄임으로서, 잘못 구성된 문제로부터 잘 구성된 문제에게로 작업을 전환한다. 인간 디자이너는 그의 장기 메모리로부터 어떤 제약들과 부목표들, 그리고 작성기들을 불러낸다. 이러한 것들은 문제 서술에서 구체화되지 않은 요소들이다. 그러므로 작업의 형성은 동적이다. 디자인 문제의 형성은 그것 자체로 문제해결작업으로 여겨질 수 있고, 과정이 계속되면서 재형성을 착수할 것이다.¹²⁾

4-3. 문제해결방식의 종류

최근에 카네기 멜론대학의 공학디자인 연구센터는 디자인에서의 형태/기능 종합을 위하여 문제해결건축의 일반적인 분류를 만들었다.¹³⁾ 그 분류는 다음과 같이 세 가지의 기본적인 전략으로 나누어진다.

- a) Top-Down 전략 : 이 전략은 여러 단계의 추상화를 통해서 디자인 명세를 발전시키는 방법이다.
- b) Bottom-Up 전략 : 이 전략은 작은 단계들로 점증적으로 디자인을 구축하는 방법이다.
- c) Middle-Out 전략 : 이 전략은 고도로 구조화된 묘사로 시작하여 주어진 요구조건들을 만족시키기 위해 그것을 변형시키는 방법이다.

이러한 각각의 전략은 기능(행위)유도접근방법(function-driven approach)과 형태유도접근방법(form-driven approach)으로 구별된다. 이러한 구별은 인공지능에서의 목표유도접근방식(goal-driven approach)과 데이터유도접근방식(data-driven approach) 사이에서 이루어진 구별과 일치한다.

1) Top-Down 전략

· 기능유도전략

MICON은 디자인하고 구축하며 실험하는 프로그램들의 통합된 집합이다.¹⁴⁾ 이 프로그램은 사용자의 고단계의 기능적인 명세를 실행하는 논리적 디자인을 생성시키기 위해 첫 단계부터 기능유도, Top-Down 전략을 이용한다.

· 형태유도 전략

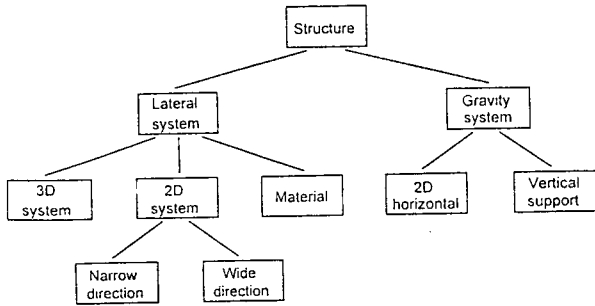
EDESYN shell은 부분들의 조립체를 물리적 구성원들과 부구성원들의 체계로 계층적으로 분해하는 것에 기반을 둔 형태유도, Top

12)Coyne, R. D., M. A. Rosenman, A. D. Radford, M. Balachandran, and J. S. Gero. 앞의 책, pp.16-17.

13)Flemming, U., J. Adams, and R. Woodbury. Computational Models for Form-Function Synthesis in Engineering Design, Report EDRC48-92, Carnegie Mellon University, Pennsylvania, 1992

14)Birmingham, W. P., and D. P. Siewiorek. "Single Board Computer Synthesis," in Expert Systems for Engineering Design, M. D. Rychener (ed.), Academic Press, New York, pp.113-139.

-Down 디자인 전략을 실행하는데 이용된다. 이러한 접근방식은 1960년대의 디자인 방법론 운동에서 전개된 "형태학적 상자"나 결정 분야의 접근방법을 연상시키고 있다. 아래의 <그림 4>는 메이어에 의한 빌딩구조의 간단한 분해를 보여주고 있다.¹⁵⁾



<그림 4> 메이어에 의한 빌딩구조의 계층적 분해

2) Bottom-Up 전략

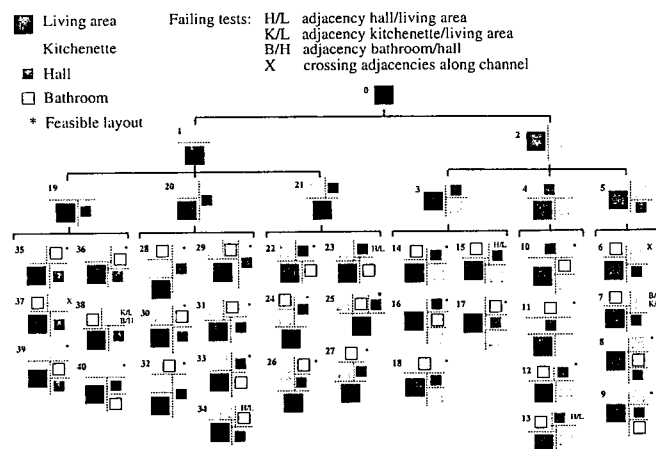
· 기능유도 전략

기능유도, Bottom-Up 전략은 개별적인 함수나 행위, 또는 성능 명세들 등을 한번에 하나씩 취급하고, 이러한 명세들을 연속적으로 고려함으로써 점진적으로 디자인의 묘사를 유도해 낸다.

이러한 전략을 사용하는 예로서 WRIGHT 시스템이 있다.

· 형태유도 전략

형태유도, Bottom-Up 전략은 소단계로 디자인을 생성하고, 과정을 약속한 방향으로 향하게 하기 위해 중간적인 평가를 이용하는 생성-실험 접근방식에 의해 사용된다. LOOS 시스템은 이러한 전략의 사례이다. 아래의 <그림 5>는 플레밍에 의해 프로그램된 것으로서, 소규모 방들의 배치 문제를 해결하기 위해 LOOS 시스템에 의해서 탐색된 상태공간이다.¹⁶⁾



<그림 5> 4개의 방을 배치하는 문제를 위해서 LOOS 시스템에 의해 생성된 상태공간

3) Middle-Out 전략

· 기능유도 전략

기능유도, Middle-Out 전략에서 시스템은 디자인의 요구되는 형

15) Maher, M. L. "Engineering Design Synthesis: A Domain Independent Representation," *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1987, 1:207-213.

16) Flemming, U. "Knowledge Representation and Acquisition in the LOOS System," *Building and Environment*, 1990, 25:209-219.

위에 대한 고도로 구조화된 묘사로 시작하고, 이러한 묘사를 물리적인 묘사로 전환한다. 그 예로서 SAW가 있다.¹⁷⁾

· 형태유도 전략

형태유도, Middle-Out 전략은 고도로 구조화된 묘사로 시작하여, 그것을 정제하거나 주어진 문맥에 적용시킨다. 격기반이나 원형기반의 접근방식이 이러한 그룹에 속한다.¹⁸⁾

추론자가 현재의 것과 비슷한 이전의 상황들을 기억하고 현재의 문제를 해결하기 위해서 그것들을 이용할 때, 인공지능 연구자들은 격기반이나 유추적 추론을 이야기한다. 이것은 바로 새로운 요구들을 만나기 위해 과거의 해결책들을 변형시키거나, 새로운 상황들을 설명하기 위해 과거의 사례들을 이용하고, 새로운 해결들을 평가하기 위해 과거의 사례를 이용하며, 새로운 상황을 해석하기 위해 선례들로부터 추론하는 것을 의미한다.¹⁹⁾

5. 현재의 미개척 영역

인공지능 및 디자인 연구 분야에는 디자인 학습 시스템과 창조적인 디자인이라는 두 개의 활동적인 미개척 영역이 있다.

5-1. 디자인에서의 기계학습

컴퓨터 프로그램이 자신의 성능을 개선시키기 위해 그 자신의 작동으로부터 배울 때 기계학습이 일어난다. 디자이너들은 자신의 활동들로부터 배우고, 그들이 가지고 있는 전문지식이라고 불리는 것은 대부분 그러한 학습의 결과로 형성된 것이다. 디자인에 필요한 지식기반은 풍성해져가고 있으며, 기계의 이용에 필요한 지식을 획득하는 문제들은, 만일 컴퓨터가 자기 자신의 작동으로부터 학습함으로써 이러한 과정에 공헌할 수 있다면 개선될 수 있다고 오래 전부터 인식되어왔다. MICON 시스템의 학습 모듈은 이러한 기능을 수행한다. 즉 그 시스템은 자신의 지식기반이 불완전 할 때 사용자에게 누락된 정보를 공급할 것을 요구하고, 전문가에 의해서 이용될 수 있는 지식획득 도구를 제공한다. 통(Tong, C.)은 Top-Down 디자인 시스템의 구문으로 유사한 접근 방식을 묘사하고 있다. BRIDGER 시스템은 Top-Down 디자인 시스템을 위한 좀더 자동화된 형식의 학습을 설명하고 있다.²⁰⁾ 이 시스템은 종합과 미학적인 위계에 기초하여 형태-유도 전략을 따르므로써 케이블로 교량을 디자인하고 있다. 브로더웨이(Braudaway)와 통(Tong)은 제어작성기(점증적인 발생-실험 접근방법의 문맥 속에서 조건을 만족시키는 부분적인 해결들을 생산하는 작성기)를 구축하는 "지식축적" 기계를 제안했다. 로젠만(Rosenman, M.A.), 지로(J.S. Gero), 그리고 옥스만(R.E. Oxman)은 "격에는 무엇이 있는가: 디자인에 있어서 격기반, 지식기반, 그리고 데이터 베이스의 사용"이라는 논문에서 격기반디자인 생성의 잠재적인 학습을 강조하고 있다.²¹⁾

17) Thomas, D. E., and R. L. Blackburn. *Algorithmic and Register-Transfer Level Synthesis: The System Architect's Workbench*, Kluwer Academic Publishers, Boston, 1990.

18) Kolodner, J. L. "Improving Human Decision Making through Case-Based Decision Aiding," *AI Magazine*, 12(2):52-68.

19) Flemming, U. 앞의 책, pp.15-16.

20) Reich, Y., R. Coyne, A. Modi, D. Steier, and E. Subrahmanian. "Learning in Design: An EDRC (USA) Perspective," in *Artificial Intelligence in Design '91*, J. S. Gero (ed.), Kluwer Academic Publishers, Boston, 1991, pp.303-321.

21) Rosenman, M. A., J. S. Gero, and R. E. Oxman. "What's in a Case: The Use of Case Bases, Knowledge Base, and Databases in Design," in *CAAD Future '91*, G. N. Schmitt (ed.), Vieweg, Wiesbaden, German, 1991, pp.285-300.

디자인에서의 기계학습에 관한 연구는 지금까지는 아직 실험적인 결과만을 내놓고 있다. 실용적으로 성공적인 상호작용의 학습모델과 Top-Down 시스템과 관련하여 발전되어온 지식획득 도구들은 엄격한 의미에서 기계학습을 표현하는 것은 아니다. 왜냐하면 인간이 활동적으로 참여하기 때문이고, 그러한 기계는 자신의 노력을 통해서라기 보다는 반복에 의해서 학습하기 때문이다.

보다 폭넓은 관심을 받기 시작한 기계학습 기술의 독특한 적용은 적응적인 인터페이스에 집중되어 있다. 적응적인 사용자 인터페이스는 사용자들의 요구에 인터페이스를 적응시킴으로써 사용자들을 지원하거나 자동적으로 적응을 수행하는 것을 말한다. 쿨(Kuhme, T.)과 그의 동료들은 CAD 시스템이 자동적으로 사용자의 전문기술의 단계를 인식할 수 있는 인터페이스를 가능하게 하는 신경망기반 접근방법을 제안했다. 이러한 인터페이스는 결과로써 생기는 분류를 기반으로 하여 사용자의 요구에 동적으로 자신을 적응할 것이다.

5-2. 창조적 디자인

창조적인 해결을 생성시킬 수 있는 컴퓨터 프로그램을 개발하는 것은 현재의 인공지능과 디자인 분야에서 궁극적인 도전으로서 나타난다. 창조성은 이러한 연관성에 의해 다음과 같은 두 가지 방법으로 정의된다. 첫째, 적용되는 방법이나 과정에 의한 정의로써, 이러한 경우의 정의는 과정들과 방법들 - 창조적 디자인을 지식근원이나 문제해결 전략이 알려져 있지 않는 곳에서의 과정으로 특정지움 - 에 의해서 주어졌다. 둘째, 디자인 과정에서 생성된 결과들에 의해 표현된 혁신의 정도에 의한 정의이다.

전산응용 창조적 설계를 향한 현재의 접근방법에는 두 개의 일반적인 경향을 가지고 있다. 첫 번째의 접근방법은 과정 자체에 집중하고 유추나 전환에 의한 추론을 선호하는 인간의 창조성에 관한 이론들에 의해 고무된 방법이다. 두 번째 접근방법은 제1근본 원리들(first principles)과 물리적인 인과관계 등을 추론하고, 인간의 창조성을 흉내내려고 하지 않고 혁신적인 해결들을 생성하고자 하는 방법이다. 자오(Zhao, F.)와 메이허(M. L. Maher)에 의해 묘사된 시스템은 첫 번째 접근방법에 속한다.²²⁾ 이 시스템은 유추기구(새로운 문맥에서 유망한 해결원형들을 검색하는)와 변환기구(원형의 적용이 실패할 때, 다른 원형의 부분들을 포함하기 위하여 야기될 수 있는)를 포함하기 위해서 원형기반디자인 개념을 확장시키고 있다. 케이건(Cagan)과 아고기노(Agogino)는 제1근본 원리에 근거하여 혁신적인 디자인을 향한 접근방법을 묘사하고 있다.²³⁾

유추와 변환기반 시스템을 사례로 취해보자. 만일 그것들이 인간의 창조성에 대한 모형으로 보여진다면, 그것들은 경험적인 증거에 의해 그것들의 타당성을 성립할 필요가 있다. 만일 그것들이 혁신적인 해결들 - 즉, 예상할 수 없는 진정으로 새로운 것 - 을 생성시키는 것으로 의미되면, 그것들은 원형에 대한 데이터 베이스가 필요하다. 유추와 변환 기구들이 그것으로 추론할 수 있도록, 어느 한 원형에 대한 묘사를 구축하는 것은 만만치 않은 작업으로 여겨진다. 필요한 범위의 데이터 베이스가 생성될 수 있는 방법은 불명확하다.

제1근본 원리들로부터 추론하는 것은 인간 디자이너에 의해 고용된 과정들과 그들에 의해 생성된 해결들 둘 다를 무시함으로써 이러한 위

험을 피할 수 있도록 해준다. 문헌을 통해 알려진 그 시스템은 증가적인 생성 - 실험의 형식을 선택하고 있으며, 이러한 접근방법에 관련된 조합적인 문제들을 직면하고 있다. 그러나 만일 창조적인 디자인이 절차적으로 한정된다면, 이러한 시스템들은 첫 번째의 단계에서 창조적 이다라고 불리어질 수 없을 것이다. 왜냐하면 그것들이 사용하는 연산자는 그것들이 생성하는 해결들이 얼마나 새로운 가에는 관계없이 처음부터 알려져 있었기 때문이다. 미첼(Mitchell, W. J.)은 창조성에 대한 유용한 정의를 찾기가 어렵다는 것을 적절히 요약하고 있다.²⁴⁾ 그는 창조적인 디자인을 설명되지 않는 범주로 특성화하고 있다. 이것은 역설적이다. 즉, 창조적인 디자인 행위의 기술을 묘사하려고 하는 그 어떠한 시도도 그 행위를 비창조적인 것으로 재정의하는 즉각적인 결과를 가져올 것이다.²⁵⁾

6. 결론

지금까지 디자이너가 가지고 있는 전문적인 지식을 어떻게 형식화하여 컴퓨터내에 표현할 수 있는지에 대한 지식기반디자인시스템의 이론적 기반을 다지기 위하여, 인공지능이나 지식공학에서 이루어진 성과들을 디자인 지식에 응용하여 보았다.

디자인을 위한 지식기반디자인시스템을 구축하기 위한 모형의 기초는 표현, 추론, 그리고 구문 등에 의해 형성된다. 이러한 지식표현안들로서 규칙과 생성시스템, 논리적 프로그래밍, 프레임과 객체지향 프로그래밍 등을 검토해 보았다. 그리고 디자인 과정을 문제해결의 과정으로 해석하여 그 방식을 검토해 보았다. 이러한 분석을 통해서 볼 때 디자인도 지식기반시스템을 적용하여 컴퓨터내에서 표현될 수 있는 가능성을 확인해 보았다. 그러나 좀 더 체계적인 디자인 요소들과 그 요소들을 구성시키는 규칙 또는 방법론에 대한 연구가 차후에 진행되어야 할 것이다.

또한 디자인에는 때때로 불합리적이고 쉽게 전산화될 수 없는 디자인 과정 즉, 비연역적인 것들이 있다. 그것은 바로 디자인의 인식적인 양상을 포함하는 디자인의 학습과 창조성에 관한 문제들이다. 따라서 진정한 의미에서 인공지능의 개념을 적용시킨 성공적이고 지능적인 지식기반시스템을 구축하기 위해서는, 시스템 자체가 입력된 데이터에 따라서 스스로 학습할 수 있어야 하고, 창조적인 디자인을 해결할 수 있는 영역에도 많은 연구가 있어야 할 것이다.

참고문헌

1. 김 현숙, 인공지능이란 무엇인가, 크라운출판사, 1995.
2. 황중선, 김 현숙 공저, 지식공학, 홍릉과학출판사, 1994.
3. G. Carrara and Y.E. Kaly, editors. Knowledge-Based Computer Aided Architectural Design, Elsevier, 1994.
4. Coyne, R. D., M. A. Rosenman, A. D. Radford, M. Balachandran, and J. S. Gero. Knowledge-Based Design Systems, Addison-Wesley, 1990.
5. Hayes-Roth, F. "Rule-Based System," Comm. of the ACM, 1985.
6. Davis, R., and J. King. "An Overview of Production Systems," in Machine Intelligence 8, Halstead Press, New York, 1977.
7. Flemming, U., R. Coyne, T. Glavin, and M. Rychener. "A Generative Expert System for the Design of Building Layout-Version 2," in Artificial Intelligence in Engineering: Design, J. S.

24) Mitchell, W. J.. "A Computational View of Creativity" in Modeling Creativity and Knowledge-Based Creative Design, J. S. Gero and M. L. Maher (eds.), Lawrence Erlbaum, Hillsdale, New Jersey, 1990, pp.25-42.

25) Flemming, U. 앞의 책, pp.20-21.

22) Zhao, F., and M. L. Maher. "Using Network-Based Prototypes to Support Creative Design by Analogy and Mutation," in Artificial Intelligence in Design '92, J. S. Gero (ed.), Kluwer Academic Publisher, Boston, 1992, pp.773-793.

23) Cagan, J., and A. M. Agogino. "Innovative Design of Mechanical Structures from First Principles," Artificial Intelligence in Engineering Design, Analysis and Manufacturing, 1987, 1:169-189.

- Gero (ed.), Elsevier, New York, 1988.
8. Flemming, U. "Artificial Intelligence and Design : a Mid-Term Review" in Knowledge-Based Computer-Aided Architectural Design, Elsevier, 1994.
 9. Flemming, U., J. Adams, and R. Woodbury. Computational Models for Form-Function Synthesis in Engineering Design, Report EDRC48-92, Carnegie Mellon University, Pennsylvania, 1992.
 10. Maher, M. L. "Engineering Design Synthesis: A Domain Independent Representation," Artificial Intelligence in Engineering Design, Analysis and Manufacturing, 1987, 1.
 11. Flemming, U. "Knowledge Representation and Acquisition in the LOOS System," Building and Environment, 1990, 25.
 12. Kolodner, J. L. "Improving Human Decision Making through Case-Based Decision Aiding," AI Magazine, 1991, 12(2).
 13. Thomas, D. E., and R. L. Blackburn. Algorithmic and Register-Transfer Level Synthesis: The System Architect's Workbench, Kluwer Academic Publishers, Boston, 1990.
 14. Pohl, J., L. Myers, J. Chapman, J. Snyder, H. Chauvet and D. Johnson. "ICADS Working Model Version 2 and Future Directions," Technical Report, CADRU-05-91, CAD Research Center, College of Architecture and Environmental Design, 1991.
 15. Gero, J. S. "Design Prototypes: A Knowledge Representation Schema for Design," AI Magazine, 1990, 11 (4).
 16. Reich, Y., R. Coyne, A. Modi, D. Steier, and E. Subrahmanian. "Learning in Design: An EDRC (USA) Perspective," in Artificial Intelligence in Design '91, J. S. Gero (ed.), Kluwer Academic Publishers, Boston, 1991.
 17. Rosenman, M. A., J. S. Gero, and R. E. Oxman. "What's in a Case: The Use of Case Bases, Knowledge Base, and Databases in Design," in CAAD Future '91, G. N. Schmitt (ed.), Vieweg, Wiesbaden, German, 1991.
 18. Zhao, F., and M. L. Maher. "Using Network-Based Prototypes to Support Creative Design by Analogy and Mutation," in Artificial Intelligence in Design '92, J. S. Gero (ed.), Kluwer Academic Publisher, Boston, 1992.
 19. Cagan, J., and A. M. Agogino. "Innovative Design of Mechanical Structures from First Principles," Artificial Intelligence in Engineering Design, Analysis and Manufacturing, 1987.
 20. Mitchell, W. J.. "A Computational View of Creativity" in Modeling Creativity and Knowledge-Based Creative Design, J. S. Gero and M. L. Maher (eds.), Lawrence Erlbaum, Hillsdale, New Jersey, 1990.

〈 접수 : 1996. 2. 29〉