

에시에 의한 프로그램 생성(PBGD) 시스템의 설계 및 구현

서 정혁, 한 현구

한국의외국어대학교 컴퓨터공학과

Design and Implementation of a Programming by Graphical Demonstration(PBGD) System

Junghuck Suh, Hyungoo Han

Dept. of Computer Science & Engineering, Hankuk Univ. of Foreign Studies

요 약

급속도로 발전한 컴퓨터 환경하에서 사용자는 훨씬 더 편리하고 손쉽게 컴퓨터를 다룰 수 있게 되었다. 그러나 이와 대조적으로 지금까지의 프로그램 개발과정은 프로그래밍 언어의 숙달을 요구함으로써 전문적인 교육을 받은 소수의 프로그래머만이 할 수 있는 분야가 되었다. 이에 본 논문에서는 직관적인 사용자 인터페이스를 기반으로 하여 프로그래밍 언어를 모르는 사람들도 자신의 필요에 적합한 프로그램을 직접 작성하여 활용하는데 도움이 되는 시스템의 개발에 필요한 기반기술을 제시하고 구현된 시스템의 전체적인 윤곽을 제시한다.

1. 서론

최근 발달된 컴퓨터의 성능과 Graphical User Interface(GUI)기반의 운영체제의 영향으로 사용자는 보다 편하고 쉽게 컴퓨터를 다룰 수 있게 되었다. 예전의 Text 기반의 운영체제에서는 사용자가 수 많은 명령어들을 숙지하여 키보드로 입력하여야 하므로 일반 사용자들은 컴퓨터란 소수의 전문가들만 사용할 수 있는 것이라고 생각하였다. 그러나 현재는 일반사용자도 쉽게 컴퓨터를 활용하여 자신의 작업능률을 향상시킬 수 있게 되었다. 현재 개발되는 프로그램들은 모두 윈도우즈 기반의 프로그램들이다. 이러한 윈도우즈 기반의 프로그램들은 사용자에게 편리함을 제공해 주고 있다. 그러나 지금까지의 프로그램 개발과정에는 프로그래밍 언어에 따른 여러 가지 규칙과 수 많은 함수들을 암기해야 하는 데 윈도우즈 프로그래밍 환경에서는 이러한 것들이 더욱 늘어났다. 이는 일반인들이 자신의 목적에 적합한 간단한 프로그램을 작성하는 것도 불가능하게 만들었으며 오직 소수의 프로그래머들만이 프로그래밍을 할 수 있도록 만들었다.

점차 다양화, 전문화 되고 있는 사회 흐름 속에 특정분야의 전문지식이 요구되는 프로그램들의 개발 요구가 증대 되고 있다. 본 논문에서는 이러한 요구를 수용하여 직관적인 사용자 인터페이스를 기반으로 프로그래밍 언어에 대하여 전혀 모르는 사람들도 자신의 필요에 적합한 프로그램을 직접 작성하여 활용할 수 있도록 도와주는 도구를 개발하는데 중점을 두었다.

본 논문은 제 2 절에서 기존의 관련 기술 현황과 문제점에 대해 기술하고, 제 3 절에서 PBGD 시스템의 기본 특징 및 동작과정에 대하여 설명한다. 마지막으로 제 4 절에서는 결론 및 향후 연구방향에 대하여 제시한다.

2. 관련연구

본 논문에서 구현한 시스템에서 가장 핵심을 이루는 기술 중 하나는 사용자가 입력한 자료를 분석하여 자동으로 규칙을 유추해내는 기술이다. 이러한 기술은 MS 워드와 같은 워드프로세서나 익스플로러와 같은 인터넷 프로그램에서 사용자가 이전에 입력한 단어와 동일한 철자들이 입력되면 이전에 입력한 단어를 미리 제시함으로써 사용자의 편의를 도모한 것과 같이 다양하게 구현되었다. 이 기법들은 워드프로세서등에서 사용되고 있는 매크로 기능과 유사하다. 그러나 매크로기능이 사용자 자신이 규칙을 설정하고 이를 활용하는데 비해 시스템이 자동으로 사용자의 행위를 규칙화하고 이를 활용하여 사용자가 좀 더 쉽게 자신의 작업을 수행할 수 있도록 도와준다.

최근에는 단순히 사용자들의 기존작업을 분석하여 새로운 작업을 예측하는 수준을 벗어나 좀더 발전된 형태로 규칙을 생성시킬 수 있는 다양한 기법들이 개발되어지고 있다. 대표적인 예로 APPLE 기반하에 작동하는 PYGMALION[4]과 그림 1에 보이는 TINKER[5]라는 프로그램이 있다.

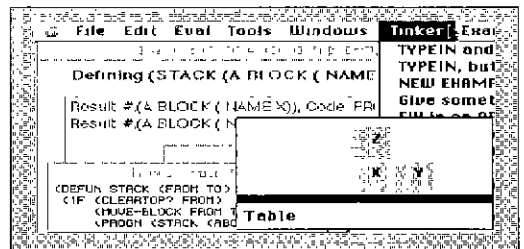


그림 1 TINKER의 실행화면

특히 TINKER는 마우스를 사용하여 사용자가 규칙을 표현하고 자체적으로 내장된 편집기를 사용하여 LISP 형태로 규칙들을 서술하면 입력된 규칙들이 실행된다

그러나 이 프로그램에서는 일반적으로 프로그래밍 언어를 사용하여 작성하는 지금까지의 프로그래밍 방식과 마찬가지로 사용자가 규칙서술의 방법을 이해하고 숙지해야만 한다

프로그래밍을 자동으로 생성하는 자동 프로그래밍 시스템의 연구가 많이 진행되어 왔다[1, 2, 3]. 자동 프로그래밍이란 인공지능[7]의 한 응용분야로서 현재 인간이 하고 있는 프로그래밍 행위의 일부를 대신 해 주어서 인간에게 보다 편리한 프로그래밍 환경을 제공해 주는 것이다

그림 2는 자동 프로그래밍 시스템의 전반적인 과정을 나타낸다.

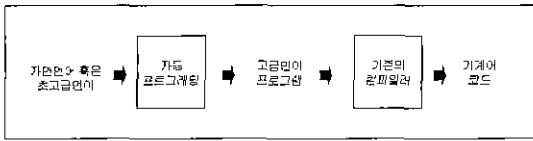


그림 2 자동 프로그래밍 시스템에 의한 프로그래밍

자동 프로그래밍 시스템은 파스칼이나 C 등의 고급언어에 의한 프로그램이 아니라 자연언어나 4세대 언어 수준인 초고급언어(Very High Level Language)명세를 입력으로 받는다 출력으로는 고급언어 수준의 목적언어 프로그램을 생성해 낸다. 생성된 프로그램은 기존의 컴파일러등에 의해서 처리되도록 구성된 시스템이다

자동 프로그래밍 시스템의 구성은 일반적으로 입력인 프로그램 명세, 출력인 목적언어, 전체 제어 구조의 운용방법, 그리고 자동 프로그래밍 적용 영역인 문제 영역으로 구성된다

자동 프로그래밍 시스템의 대표적인 예로는 스탠포드 대학에서 만든 PSI[1]시스템이 있으며 이것은 자동 프로그래밍의 여러 기법을 종합적으로 사용하여 구현한 방대한 크기의 자동 프로그래밍 시스템이다 또한 국내에서 제작된 시스템인 KAPS[1]는 지식기반 자동 프로그래밍 시스템으로서 입력으로 초고급언어에 해당하는 추상알고리즘과 자료형을 받아들여서 출력으로 C 프로그램을 생성해 낸다

자동 프로그래밍 시스템들은 사용자에게 편리함을 주는 데 목적을 가지고 있으나 일반인들이 아닌 전문적인 교육을 받은 프로그래머를 대상으로 하고 있다 따라서 일반인들이 쉽게 자신의 목적에 적합한 프로그램을 만들 수 있는 직관적인 사용자 인터페이스를 제공하지 못하였다

3. 시스템 구성

2절에서 언급한 여러가지 단점을 극복하기 위해 본 연구에서는 다음과 같은 전체사항을 두었다. 첫째 TINKER의 경우처럼 사용자가 LISP 과 같은 언어를 사용하는 것이 아니고 직관적인 그래픽 인터페이스를 사용자에게 제공하여 단순히 인터

페이스상의 아이콘들을 선택함으로써 자신의 규칙들을 생성하도록 하였다. 또한 일반인들을 대상으로 한 것이기 때문에 고급언어로 된 프로그램 코드를 작성하는 대신 시스템 내부에서 자체적인 가상코드를 구성하여 실행하도록 하였다.

본 연구에서 개발한 PBGD 시스템은 윈도우 95 환경에서 작동하도록 작성되었으며 개발도구는 파스칼언어를 기초로 한 델파이 3.0으로 구현하였다.

전체적인 시스템의 구성도를 살펴보면 그림 3과 같다.

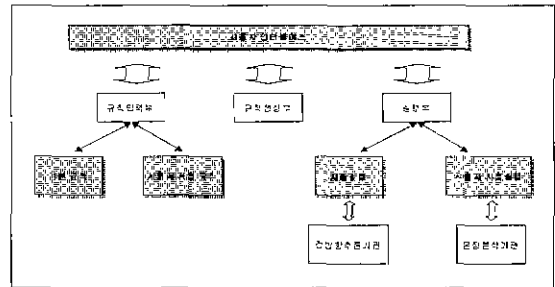


그림 3. 시스템 구성도

사용자는 셀방식의 규칙편집기에서 각 OBJECT에 대한 규칙들을 아이콘형태의 그래픽으로 입력한다 규칙생성기에서는 규칙입력부에서 입력받은 그래픽들의 변화과정을 분석하여 규칙들을 자동 생성한다. 실행부는 규칙생성부에서 생성된 규칙들을 활용하여 프로그램을 수행한다

사용자는 자신이 원하는 규칙에 따라서 초기, 중간, 결과의 3단계의 그래픽을 입력할 수 있고 그 중에서 중간 단계는 생략이 가능하다. 또한 사용자는 자신의 편의에 따라 시스템에서 순서적으로 지정하는 기본규칙과 사용자 자신이 직접 규칙의 이름을 지정할 수 있는 두 가지 방법 중에서 선택할 수 있다

그림 4는 사용자가 규칙을 입력하는 규칙편집기에서의 작업 화면으로 스머프가 가가벨에게 잡아 먹히는 것을 프로그램 하는 과정을 보이고 있다

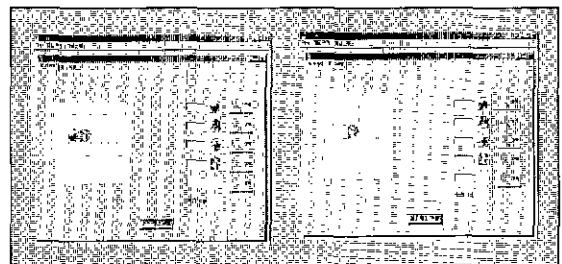


그림 4. 사용자 규칙입력부분

왼쪽의 초기화면에서는 스머프 OBJECT와 가가벨 OBJECT가 서로 만나고 오른쪽의 결과화면에서는 스머프 OBJECT는

사라지고 가가멜 OBJECT 만 남도록 함으로서 스머프가 가가멜에게 잡아 먹히는 것을 그래픽으로 표현하던 그에 대한 프로그램이 자동 생성된다.

이와 같은 방식으로 사용자는 OBJECT 가 옆으로 움직이거나, 장애물을 뛰어 넘는 것, 시간의 흐름에 따라 새로 태어나고, 성장하며 죽는 것 등의 다양한 형태의 행동들을 손쉽게 프로그래밍화 할 수 있다.

규칙생성부에서 생성된 각각의 규칙들은 그 규칙이 발생할 수 있는 조건을 나타내는 조건부와 그러한 조건이 만족되었을 때 수행되는 실행부로 구성된다.

실제 실행 단계에서는 다음과 같은 두 가지의 방식 중에서 사용자가 하나를 선택하여 작성된 프로그램을 실행할 수 있도록 하였다.

가) 자동실행방식

사용자가 입력한 규칙들을 이용하여 시스템에서 자동으로 실행하는 방식이다. 이 방식에는 인공지능의 한 분야인 전문가 시스템에서 사용되는 정방향 추론기법[6]을 사용하여 수행되도록 하였다.

정방향 추론은 역방향 추론과 더불어 전문가 시스템의 주요 추론 방법중의 하나이다 일반적으로 정방향 추론은 패턴매칭(Pattern-Matching), 상충 해소(Conflict Resolution), 수행(Firing)의 3 단계를 반복 수행하게 된다

자동실행 방식은 먼저 패턴매칭을 통해 현재 상태에서 적용될 수 있는 규칙들을 찾아낸다 만약 수행될 수 있는 규칙들이 여러 개일 경우 상충해소기법을 사용하여 그 중의 하나를 선택한다. 선택된 규칙의 실행부에 나타난 방법대로 실행하고 실행이 종료된 뒤 그 실행에 의하여 변화된 상태를 반영하여 위의 과정을 반복하도록 하였다

나) 사용자 지정 방식

사용자가 실행과정을 직접 기술하는 방식이다 이 방식은 사용자가 미리 지정한 각 OBJECT 들의 이름과 규칙의 이름을 사용하여 순서에 맞게 선택한 후 실행하는 방식이다 이 방식은 2 결에서 언급된 TINKER 의 방법과 유사하니 TINKER 에서는 프로그램 언어의 형태로 실행과정을 기술해야 했다 그러나 본 시스템은 사용자가 작성한 OBJECT 들과 규칙 들을 사용자에게 제시하고 그 중에서 자신이 필요한 것들을 마우스로 선택하여 프로그램 구성창으로 적절하게 끌어 넣음으로써 사용자의 의도대로 프로그램을 작성하여 실행 할 수 있도록 하였다

4. 결론 및 향후 연구방향

현재 사용자가 쉽게 자신의 프로그램을 작성할 수 있도록 도와주는 시스템들은 여러 분야에서 개발되고 있다 예를 들면 Rapid Application Development(RAD)틀이라고 불리는 델파이나 비주얼베이직 등의 개발 툴 들과 자동 프로그래밍 시스템들과 같이 손쉽게 프로그램을 작성할 수 있도록 도와주는

시스템들이 있다 그러나 이러한 시스템들은 그 시스템의 사용자를 이미 프로그램 언어나 프로그램 기법에 대하여 익숙한 사람들을 대상으로 하고 있다. 따라서 프로그램의 구현 원리를 전혀 알지 못하는 일반인에게는 도움이 되지 못한다 본 연구에서는 사용자가 자신의 논리를 쉽게 프로그램으로 표현함으로써 보다 적극적이고 논리적인 사고력 향상 및 프로그램에 대한 흥미를 느낄 수 있도록 직관적인 사용자 인터페이스를 제공하는 시스템을 개발하였다.

본 시스템은 사용자들이 일련의 규칙들을 잘못 입력하였을 경우에는 논리적인 규칙을 생성하지 못한다 사용자에게 의하여 입력된 일련의 규칙들의 논리적 타당성을 파악하는 연구가 계속 되어야 할 것이다 또한 셀 방식의 2 차원 사용자 인터페이스를 사용하였으나 향후 Virtual Reality Modeling Language (VRML)[8] 등의 기법을 사용하여 3 차원 사용자 인터페이스를 구현 할 계획이다 3 차원 인터페이스를 제공하면 사용자는 더욱 현실성 있는 프로그램을 작성 할 수 있으며 OBJECT 의 움직임을 보다 자연스럽게 보여줄 수 있다

본 시스템은 또한 prototype 으로서 프로그램 작성 환경을 초등학교 저학년 어린이들을 대상으로 하였다 향후 연구에서는 보다 더 실용적이고 모든 일반인들이 활용할 수 있을 정도의 수준 높은 프로그램 작성 환경을 제공하도록 하여야 할 것이며 생성된 목적 프로그램이 일반 프로그래밍 언어로 작성된 프로그램에 삽입되어 실행 될 수 있도록 모듈화된 실행 화일을 생성하는 방법을 연구하여야 할 것이다.

5. 참고문헌

- [1] 정동목, 김영택, “지식기반 지능 프로그래밍 시스템 KAPS 의 설계와 구현”, 1987
- [2] 박경환, 김영택, “논리적 구문에 대한 규칙기반 프로그램 구성”, 1988
- [3] 이강태, 남광우, 류근호 “지식베이스의 객체 지향 데이터 모델링”, 정보과학회 학술발표논문집(B), 1998
- [4] David Camfield Smith “Pygmalion An Executable Electronic Blackboard” In Watch what I do, edited by A. Cypher, pp 19-48 MIT Press, Cambridge MA. 1993
- [5] Henry Lieberman “Tinker A Programming by Demonstration System for Beginning Programmers” In Watch what I do, edited by A Cypher, pp 19-48 MIT Press. Cambridge MA 1993
- [6] 이재규, 최형림, 김현수, 서민수, 주석진, 지원철, “전문가 시스템 원리와 개발”, 1996
- [7] 김성식, “인공지능 기법”, 1996
- [8] VRML97 Specification. “http //www vrml org/Specification/VRML97”. 1997