

공장자동화를 위한 지식처리형 작업계획 시스템 도구의 개발*

강희중, 김진형

한국과학기술원, 전산학과

Development of Knowledge-based Planning and Scheduling System Shell for Factory Automation

Heejoong Kang, Jin H. Kim

KAIST, Department of Computer Science

요약

계속적으로 발생하는 주문으로부터, 임의의 주문을 수행하기 위해 작업계획을 수립하는 문제는 그 문제가 지닌 시간적 복잡도와 실행적 불확실성 때문에 매우 어려운 문제로 알려져 있다. 그러나, 공장자동화를 이루기 위해서 컴퓨터에 의한 작업계획 생성 및 제어의 필요성은 점차 증대되고 있으며, 그에 따라 작업계획 전문가시스템도 많이 개발되고 있다. 본 논문에서는, 그러한 작업계획 전문가시스템을 쉽게 개발할 수 있도록 인공지능 기법인 지식처리형 기법으로 지식베이스, 작업계획 문제 해결기(Planner and Scheduler), 추론 기관, 그리고 윈도우와 그래픽 기반의 사용자 인터페이스로 구성된 지식처리형 작업계획 시스템 도구를 개발하여, 모형적 지식처리형 작업계획 시스템을 구현하고, 작업계획 전문가 시스템 개발 환경을 구축하고자 하였다.

I. 서론

고객으로부터 명세된 주문의 제한조건을 만족하고 그 조건에 따라 생산활동을 제어할 수 있는 작업계획을 수립하는 것은 지수함수적인 시간적 복잡도와 수립된 작업계획을 실행하는데 있어서 존재하는 실행적 불확실성 때문에 매우 어려운 문제로 알려져 있다. 일반적으로 작업계획 문제는 각 주문에 대해서 작업의 순서를 결정하는 작업경로 선정 문제와 결정된 순서에 따라 자원과 시간을 각 작업에 할당하는 자원할당 문제로 이루어져 있다[1]. 이 두 문제는 실제로 매우 밀접한 관계를 맺고 있어서 작업계획 문제의 시간적 복잡도를 NP-Hard로 만들고, 여기에 덧붙여 공장의 실제적 생산활동에서 예측할 수 없는 상황을 고려한다면 작업계획 문제는 더욱 복잡해질 수 있다. 작업계획은 납기일, 비용, 기계의 유용성 및 대치 여부, 자재의 충족 여부 등과 같은 제한조건들이 만족되지 않고 상충될 때 고객이나 생산자의 입장에서 완화되어 수립될 수 있다. 작업계획 시스템은 작업계획을 수립하고, 예측할 수 없는 사건의 발생, 즉, 기계의 고장, 기계 조작자의 부재, 주문 명세의 변경 등에 대하여 그 사건의 영향을 흡수하여 작업계획을 재수립할 수 있어야 한다.

최근에 인공지능 기술이 발달함에 따라, 인공지능 기법을 이용하여 작업계획 문제를 해결하려는 연구가 활발히 수행되고 있다. 이러한 연구의 결과로서 ISIS, OPIS 등과 같은 많은 작업계획 전문가시스템이 개발되었다[2]. 작업계획 전문가시스템은 작업환경, 제한조건, 제한조건의 완화를 결정하는 규칙 등에 따라 달리 개발될 수 있으며, 문제 해결에 있어서 작업계획 전문가의 전문지식과 경험적 지식을 최대한 활용하여 효율적인 탐색을 수행한다. 이를 위하여 여러 공장에 적용할 수 있는 작업계획 전문가시스템 도구의 개발과 작업계획 전문가시스템 개발 환경을 구축하는 것이 필요하다.

본 논문에서는 지식베이스, 작업계획 문제 해결기, 추론 기관, 그리고 윈도우와 그래픽 기반의 사용자 인터페이스로 구성되고 지식처리형 기법으로 개발된 작업계획 시스템 도구 및 각 구성 요소들의 역할과 기능을 기술하고, 교환기 제조 공장을 위한 모형적 지식처리형 작업계획 시스템을 한 예로 구현하고자 한다. 이를 토대로 공장자동화를 위한 여러 작업계획 시스템을 구축할 수 있을 것이다[3]. 지식처리형 작업계획 시스템들의 개발과 작업계획 시스템의 구현에 복합형 인공지능 도구인 KEE(Knowledge Engineering Environment)와 인공지능 컴퓨터인 TI EXPLORER 리스프(LISP) 머신을 사용하

* 본 연구는 과학기술원 교수 자체연구 프로그램의 지원과 KAIST의 장신, 최순태씨에 의뢰된 자료제공을 받았다.

II. 작업계획 문제와 개발 환경

1 작업계획 문제

제조공장에서 작업계획 문제는 전치리로 주문의 순서를 결정하는 주문의 시이퀀싱(Sequencing) 문제, 작업 순서 결정의 플래닝(Planning) 문제, 그리고 각 작업에 자원과 시간을 할당하는 스케줄링(Scheduling) 문제로 정의될 수 있다[1,4] 여기서, 플래닝은 제한조건, 유한한 자원 및 기계, 공장의 생산능력 등에 무관하게 작업계획을 수립함을 의미하고, 스케줄링은 상충되는 제한조건의 완화 여부, 자원의 충족 여부, 기계의 유용성 등을 모두 고려하여 시간에 따르는 작업계획을 수립함을 의미한다 이러한 작업계획 문제는 지수함수적인 시간적 복잡도를 지니고 있다 즉, 10 개의 주문이 있고, 각 주문이 5 개의 작업을 거쳐야 한다면, 가능한 작업계획은 $(10!)^5$ 또는 10^{30} 이상이 되고, 더 많은 주문과 작업이 있으면 작업계획 문제는 더욱 더 복잡해진다.

본 논문에서는 작업계획 문제를 주문 시이퀀싱, 작업 플래닝, 그리고 작업 스케줄링으로 나누어 해결하려고 하였다. 이 문제의 해결은 스케줄링 전문가의 경험과 전문지식을 활용할 수 있는 지식처리형 기법에 의존하였다. 지식처리형 기법은 인공지능 기법의 한 분야로서 문제 영역에 있는 전문가의 전문 지식과 경험을 최대한 활용하여 문제를 해결하려고 한 방법이다[5] 주문 시이퀀싱 문제는 어떤 순서에 이 때까지 들어온 주문들의 순서를 결정하는 것으로, 주문의 납기일과 우선 순위와 고객이 명세한 제한조건으로 여기고, 제한조건을 완화하는 규칙과 고객에 대한 질의 응답을 통하여 명세된 제한조건들이 최대한 만족되도록 작업에 들어갈 주문의 순서를 결정한다 작업 플래닝 문제는 결정된 주문의 순서에 따라 작업계획을 수립하는 것으로, 공장의 생산능력에 관계된 제한조건, 자원의 충족 여부, 기계의 유용성 등에 제한을 받지 않고 작업계획을 수립하며, 작업계획을 수립하는 입장에 따라서 제품 생산 계획, 자재 소요 계획, 기계 가동 계획, 인력 수급 계획, 그리고 작업 수행 계획 등이 수립될 수 있다 이렇게 수립된 작업계획은 작업 스케줄링 문제에서 고려되는 모든 제한조건들이 만족될 때, 비로서 자원과 시간을 할당받아 실제 현장에서 실행될 수 있는 작업계획으로 수립되며, 제한조건의 변경이 발생하면 이미 수립된 작업계획은 재수립하거나 적당한 조건을 사용자에게 제공하게 된다 물론, 필요한 경우에는 제한조건을 규칙 또는 사용자와의 질의 응답에 의해서 완화시킬 수 있다

2 복합형 개발 시스템(KEE)과 리스프 머신(TI EXPLORER)

프로그래밍 시스템을 개발하는 사람들은 그 시스템이 다른 사람에게 쉽게 이해되고, 또 수정하기가 쉽도록 시스템을 개발해야 한다[6] 프로그래밍과 프로그램을 구현할 적합한 언어의 개념적인 모델에 기초하여 프로그램들을 조직화하는 방법을 제공하는 프로그래밍 스타일이 많이 있다 이 프로그래밍 스타일은 프로시유어, 객체, 논리, 규칙, 그리고 제한조건 등에 기초

프로그래밍 파라다임(Paradigm)을 통합하려는 움직임이 있어 왔다 다양한 파라다임들의 사용은 한 스타일 이상을 익혀야 하는 부가적인 비용을 요구하고, 프로그램은 한 스타일 내에서 정보 처리를 최적화하기 위하여 같은 정보의 다른 표현으로 서로 변환되기도 한다 그러나, 프로그래밍 언어들은 더 이상한 스타일에만 중점을 두지 않는다 즉, 다양한 스타일들이 공존하게 되었다.

프로그래밍 개발 환경은 프로그램을 구축하고 수정하며 더 버그하기 위해 사용되는 도구들의 집합체이다 인공지능 프로그래밍 개발 환경은 프로그램의 구조를 분석하고, 프로그래머가 시스템을 쉽게 개발할 수 있도록 유용한 각종 도구들을 제공한다. 이 환경은 또한 프로그램의 변경을 명세하는 단순한 수단을 제공함으로써 프로그래머의 부담을 덜어 준다 이들 도구는 특히 인공지능에서 사용되는 실험적인 프로그래밍 스타일에 중요하다. 전형적인 인공지능 시스템 개발은 프레임, 규칙, 또는 논리 프로그래밍 등과 같은 자연스럽고 효율적이며 일정한 방법론 중 한 가지 방법론만을 강조했다 다양한 프로그래밍 파라다임을 통합하려는 새로운 개발 방법론이 바로 복합형 개발 시스템이다[7] 이 개발 시스템은 빠른 모형적 시스템의 개발, 문제 기술의 집진적인 세밀화, 다양하고 통합된 해결 방법의 사용, 그리고 문제 해결 과정과 문제 영역의 명백한 기술의 가시성에 중점을 두었다 이들 복합형 개발 시스템의 장점은 자연스럽고 명백한 지식 표현, 응용성 있는 사용자와의 상호 작용, 그리고 그래픽을 이용한 강력한 설명 기능 등이다 KEE는 이러한 복합형 개발 시스템의 대표적인 한 예이다 KEE는 프레임 기반의 지식 표현, 규칙 기반의 추론, 객체 지향적 프로그래밍, 리스프, 시스템과 대화가 가능한 그래픽, 그리고 조건에 따라 행동을 유발하는 Active Values 등을 지원한다[8]

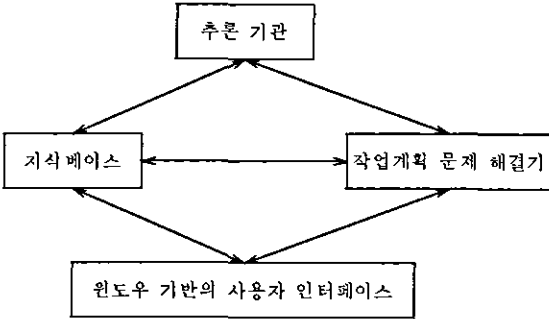
한편, 리스프 머신은 인공지능 언어인 리스프 프로그램의 수행 속도를 빠르게 하기 위하여 특별히 개발된 워크스테이션용 컴퓨터이다 컴퓨터의 모든 입력은 키보드와 마우스에 의해서 각각 행해질 수 있다. 하드웨어적인 특성을 살펴보면 특별히 고안된 리스프 프로세서가 리스프 프로그램을 빠르게 실행시키고, 함수의 순환적인 호출을 지원하기 위해 스택 구조를 지니며, 효율적인 메모리 관리를 위하여 태그(Tag)를 지닌 데이터 표현을 채택하고, 높은 해상도를 지닌 디스플레이(Display), 주 메모리와 보조 메모리 간의 고속 스위칭(Swapping) 등이 있다 소프트웨어적인 특성으로는 리스프 인터프리터, 리스프 컴파일러, 입력 에디터, 에디터, 인스펙터(Inspector), 활자체 에디터, 강력하고 다양한 윈도우 시스템 등이 있다 TI EXPLORER 리스프 머신은 위와 같은 기능들을 지원하고 있다.

III. 지식처리형 작업계획 시스템 도구의 개발

1. 구성도

지식처리형 기법으로 작업계획 문제의 해결은 물론, 다양한 작업계획 전문가시스템의 개발을 용이하게 하기 위하여 작업계획 시스템 도구의 개발은 필요하게 되었다 이 도구는 형

태상 전문가시스템의 골격과 비슷하고, 기능상 문제 영역의 지식이 없는 전문가시스템으로서 EMYCIN과 비슷하다고 볼 수 있으며, 작업계획 전문가시스템의 개발 환경 구축에 기여한다 지식처리형 작업계획 시스템 도구는 지식베이스, 작업계획 문제 해결기, 추론 기관, 그리고 사용자 인터페이스로 구성되며, 이들 각 구성 요소에 대한 관계 및 구성도는 다음과 같다



<그림 1> 지식처리형 작업계획 시스템 도구의 구성도

추론 기관은 정방향 추론과 역방향 추론이 가능하고, 규칙 기반의 추론을 행할 때 몇 가지의 제어 정책이 있어서 둘 이상 상충되는 규칙들을 해결할 수 있다 지식베이스는 크게 서술적 지식과 규칙적 지식으로 구성되고, 서술적 지식은 주로 문제 영역에 관련된 지식으로 구성되며 규칙적 지식은 시이퀀싱, 플래닝, 그리고 스케줄링에 관련된 지식으로 구성된다. 이들 지식은 모두 프레임의 형태로 지식베이스에 저장된다[8,9] 특히, 규칙적 지식은 추론의 방향과 제어 정책에 관한 정보를 함께 지니고 있으므로 사용자에게 의해서 제어될 수 있다 작업계획 문제 해결기는 주문 시이퀀서(Sequencer), 작업 플래너(Planner), 그리고 작업 스케줄러(Scheduler)로 구성된다 주문 시이퀀서는 들어온 작업들을 납기일과 우선 순위에 따라 일련의 순서로 나열하고 예상되는 주문 수행의 시작 시간과 끝 시간을 결정한다 여기서, 납기일과 우선 순위는 제한조건으로서 작용한다 작업 플래너는 모든 제한조건에 무관하게 주문을 수행하기 위한 작업계획을 수립한다 수립된 계획은 작업 스케줄러에 의하여 고객과 생산자 간의 제한조건을 모두 고려함으로써 자원과 시간을 할당받아 조정된다 작업 스케줄러의 또 다른 기능은 작업계획의 실행시에 예측할 수 없는 사건의 발생으로 작업계획의 수정이 필요할 때 작업계획을 재수립한다는 것이다 예측할 수 없는 사건은 시뮬레이션 방법에 의하여 발생되고, 그 영향은 작업계획의 재수립에 미칠 것이다 사용자 인터페이스는 강력한 윈도우와 그래픽을 제공한다 이들 윈도우와 그래픽은 사용자와의 질의 응답 또는 제한조건의 위반시 경보 상태 표현을 지원하기 위하여 사용된다 이들 각 구성 요소는 TI EXPLORER 리스프 머신 상의 KEE와 리스프 프로그래밍으로 구현되었다.

2 지식베이스와 추론 기관

지식처리형 시스템에서의 지식베이스는 매우 중요한 역할을 한다 크게 서술적 지식과 규칙적 지식으로 구성된 지식베이스는 프레임 형태로 저장되는데, 프레임은 한 객체를 기술하는 방법이다 이들 프레임들은 여러 트리를 형성하고 있다

로 간의 관계에 따라 상위 프레임의 특성이 하위 프레임에 유전되는데, 이 유전 메커니즘을 적절히 선택하면 효율적으로 지식베이스를 구축할 수 있다 서술적 지식은 주로 문제 영역을 모델링한 지식, 규칙 기반의 추론을 유도하는 지식, 사용자 인터페이스에 관련된 지식, 그리고 각종 윈도우와 그래픽에 관련된 지식 등으로 구성된다. 이들 서술적 지식은 문제 영역에 따라 수정될 수 있다 규칙적 지식은 들어온 주문의 순서를 결정하기 위한 지식, 작업계획을 수립하기 위한 지식 등으로 구성된다 규칙의 일반적인 형태는 다음과 같다.

IF <condition> THEN <conclusion> DO <action>

<condition>에는 프레임의 정보, 리스프 함수 등이 삽입될 수 있고, <conclusion>에는 프레임의 새로운 정보가, <action>에는 리스프 함수가 각각 삽입될 수 있다 즉, 이들 규칙적 지식은 서술적 지식을 이용하여 규칙 기반의 추론을 수행하면서, 서술적 지식을 변경하거나 임의 리스프 함수를 호출하여 적당한 행위를 한다. 위에서 THEN 부분 또는 DO 부분이 생략될 수 있다 즉, 정방향 추론의 경우에는 THEN 부분이, 역방향 추론의 경우에는 DO 부분이 흔히 생략된다

규칙적 지식을 이용하여 규칙 기반의 추론을 수행하는 것은 추론 기관이다 이 추론 기관은 규칙에 포함된 추론의 방향과 제어 정책에 관한 정보에 따라 규칙 기반의 추론을 수행하며, 서술적 지식을 변경시키기도 한다 작업계획과 같은 문제는 초기에 작업계획의 수립을 위해서 정방향의 추론을 수행하고, 나중에 변경된 제한조건을 만족시키기 위해서 역방향의 추론을 수행한다 대부분의 규칙들은 크게 추론의 방향에 따라 분류될 수 있으므로 프레임 형태로 저장되는 것이 유리하다 이렇게 분류된 규칙들이 규칙 기반의 추론을 행하면서 둘 이상의 상충되는 규칙이 발생하면, 지정된 제어 정책에 따라 하나의 규칙을 선택하고 규칙 기반의 추론을 계속 수행한다

3 작업계획 문제 해결기

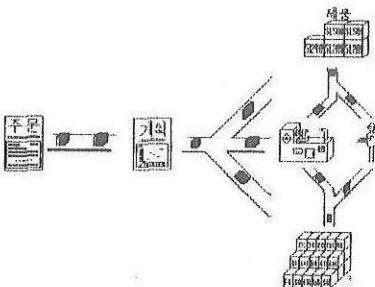
실제로 작업계획 문제를 해결하는 것은 바로 작업계획 문제 해결기이며 주문 시이퀀서, 작업 플래너, 그리고 작업 스케줄러로 구성되어 있다 이 문제 해결기에는 스케줄링 전문가의 경험적 지식이 프로그램의 형태로 포함되어 있어 가능한 작업계획의 해를 효율적으로 구할 수 있다 지식베이스, 추론 기관과 대화를 함으로써 효율적으로 가능한 해를 구하고, 윈도우 기반의 사용자 인터페이스와 대화함으로써 사용자와의 질의 응답을 편리하게 수행할 수 있다. 즉, 지식베이스와의 대화는 새로운 지식의 삽입 및 지식의 활용과 변경을 의미하고, 추론 기관과의 대화는 규칙 기반의 추론을 수행함을 의미하며, 윈도우 기반의 사용자 인터페이스와의 대화는 윈도우나 그래픽을 통한 시스템과 사용자간의 질의 응답을 의미한다. 각 구성 요소에 대해서 살펴보면 다음과 같다.

주문 시이퀀서는 현재 시간을 기준으로 이제까지 들어온 주문들을 고객이 명시한 제한조건인 납기일과 우선 순위에 맞게 일련의 순서로 배열한다. 또한 주문들의 순서가 결정된 후, 주문의 작업 실행에 관한 시작과 끝의 예정 시간을 결정할 수 있다 예정된 끝 시간이 납기일에 만족되지 않으면 납기일의

완화 또는 제품 구매의 변경 등이 필요하게 된다. 또한 사용자의 주문 입력시에 발생할 수 있는 에러에 대한 에러 복구 루틴도 필요하다. 작업 플레너는 순서대로 배열되고 예정 시간이 걸린 주문들에 대해서 플레닝을 수행하여 작업계획을 수립하는데 있어서 생산활동의 기본 단위인 주문, 작업, 자재, 기계, 제품, 인력 등의 입장에서 계획을 수립하게 되며, 이렇게 수립된 계획들은 나중에 작업 스케줄러에 의해서 사용된다. 끝으로 작업 스케줄러는 수립된 작업계획에 실제로 자원과 시간을 할당하면서, 고객과 생산자의 제한조건이 만족되는가를 검사하여 만족되도록 작업계획을 조정하거나 재수립한다. 또한 예측할 수 없는 사건의 발생이 작업계획에 어떻게 영향을 주는가를 살펴보기 위하여 시뮬레이션 방법을 사용한다. 작업 스케줄러는 이 사건에 대해 기존의 작업계획을 재수립하거나 사용자에게 유익한 조언을 제공하게 된다.

4. 윈도우 기반의 사용자 인터페이스

전문가시스템 도구의 개발에 있어서 친밀감과 융통성을 주는 사용자 인터페이스는 사용자에게 시스템과 보다 쉽게 대화할 수 있는 통로를 제공한다는 점에서 매우 중요한 역할을 하게 된다. 최근 컴퓨터 그래픽의 발달은 인공지능 개발 시스템으로 하여금 컴퓨터 그래픽과 결합하게 한다. 리스프 머신에서 제공하는 윈도우 시스템을 KEE에서 제공하는 그래픽 이미지와 결합하여 윈도우 기반의 사용자 인터페이스를 구축함으로써, 사용자는 보다 쉽고 편리하게 시스템을 사용할 수 있다. 사용자 인터페이스에서 제공하는 윈도우에는 주문의 입력, 변경된 제한 조건의 입력, 작업계획의 출력, 문제 영역 지식의 입력 및 출력 등이 있으며, 그래픽 이미지에는 각종 정보 상태 이미지, 마우스로 선택이 가능한 시스템 메뉴, 각종 정보의 상태 이미지, 그리고 작업계획의 단위를 나타내고 마우스로 선택이 가능한 아이콘(Icon) 이미지 등이 있다. 이러한 윈도우와 그래픽 이미지는 원하는 용도에 적합하게 세로이 고안되고 구현되었다. 또한 사용자 인터페이스는 전문가시스템 개발 환경을 구축하는데에도 기여하여 세로이 개발될 많은 전문가시스템 도구, 인공지능 개발 시스템 등은 윈도우 시스템 또는 컴퓨터 그래픽을 반드시 지원해야 할 것이다.



<그림 2> 작업계획 시스템 도구의 아이콘 이미지

위의 아이콘 이미지들은 KEE의 비트 맵 에디터를 이용하여 구현한 것으로 간단한 작업계획 시스템의 모델을 한눈에 보여주고 있다. 마우스에 의하여 선택된 아이콘은 메뉴를 나타내며, 메뉴가 선택되면 그에 따른 적당한 행위를 하게 된다.

VI. 결론

작업계획 문제를 완벽하게 해결하여 공장자동화에 기여하는 것은 매우 어려운 문제이다. 그러나, 본 논문에서는 작업계획 문제 해결의 실마리를 제시하고자, 인공지능 기법인 지식처리형 기법으로 작업계획 문제를 해결하려고 시도했으며, 그 결과로 지식처리형 작업계획 시스템 도구를 개발하였다. 또한 이 도구를 이용한 모형적 작업계획 시스템을 구현하고 있다. 실제로, 개발된 지식처리형 작업계획 시스템 도구가 얼마나 작업계획 전문가시스템을 개발하는데 큰 기여를 하는가를 증명할 방법은 없으나, 그러한 전문가시스템의 개발을 돕는 개발 환경을 구축하는데 큰 기여를 했다고 볼 수 있겠다. 앞으로의 과제는 보다 효율적으로 다양한 작업계획 시스템을 개발할 수 있도록 지식처리형 작업계획 시스템 도구를 보완 또는 확장하고, 다른 문제 영역에서의 작업계획 시스템을 개발함으로써 그 우수성을 증명하며, 실제 현장에서 사용될 수 있는 작업계획 시스템을 개발하는 것이다.

참고 문헌

- [1] M. S. Fox, "Constraint-Directed Search: A Case Study of Job-Shop Scheduling," Ph.D. Dissertation, CMU-RI-TR-83-22, CMU, 1983.
- [2] S. F. Smith, M. S. Fox, P. S. Ow, "Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems," *AI Mag.*, pp.45-61, Fall 1986.
- [3] D. A. Bourne, M. S. Fox, "Autonomous Manufacturing: Automating the Job-Shop," *Computer*, pp.76-86, September 1984.
- [4] M. S. Fox, S. F. Smith, "ISIS - a knowledge-based system for factory scheduling," *Expert Systems*, vol.1, no.1, pp.25-49, 1984.
- [5] F. Hayes-Roth, "Knowledge-Based Expert Systems," *Computer*, pp.263-273, October 1984.
- [6] D. G. Bobrow, M. J. Stefik, "Perspectives on Artificial Intelligence Programming," *Science*, vol.231, pp.951-957, February 1986.
- [7] J. C. Kunz, T. P. Kehler, M. D. Williams, "Applications Development using a Hybrid AI Development System," *AI Mag.*, pp.41-54, Fall 1984.
- [8] R. Fikes, T. Kehler, "The Role of Frames-Based Representation in Reasoning," *CACM*, vol.28, no.9, pp. 904-920, September 1985.
- [9] I. P. Goldstein, R. B. Roberts, "NUDGE, A Knowledge-Based Scheduling Program," *Proceedings of The Fifth IJCAI*, pp.257-263, August 1977.