

고수준 명령어로부터 명시적 로봇 프로그램 자동 유도방법*

서윤호 · 정덕호

울산대학교 수송시스템 공학부 제품 및 공정설계(PPD) 연구실

Automatic Derivation of Explicit Robot Programs from Task-Level Commands

Yoonho Seo · Deokho Cheong

Robot task program is needed to control and manage a Robot without explicitly describing the robot program by user which includes commands, procedures, geometric and signal data in the detail level. To use the Robot task program, a computer system is required to convert the Robot task into the Robot program, which can be understood by the Robot. In this paper, the systemic method for automatic generation of explicit Robot programs (ERP) from task-level commands is described. Specifically, a 3-step procedure including Robot task decomposition, task synchronization and ERP generation is presented.

1. 서론

컴퓨터를 이용한 새로운 공작기계의 개발은 제조산업에 큰 변화를 가져다 주었으며, 이러한 변화는 FMS를 이루는 기반이 되었다. 더욱이 산업용 로봇, CNC 공작기계, AGVS 등 개별적으로 지능적 운용이 가능한 구성 장비의 도입과 전체 FMS의 조직, 설계, 운영에 자주적, 지능적 요소의 추가를 확대해 나감으로써 무인자동화의 목표를 향하여 나아가고 있다. 이와 같이 제조시스템의 무인화를 위해서는 다른 지능형 장비와 더불어 산업용 로봇의 도입이 필수적인 요인으로 꼽히고 있다. 그러나 로봇 도입의 중요성에도 불구하고 산업용 로봇 시장은 예측만큼의 성장을 이루지 못하고 있다. 이러한 저성장의 가장 주된 이유 중 하나로 산업용 로봇을 전체 제조시스템과 상호 보조를 맞추며 효율적으로 활용되게 하기 위한 소프트웨어의 개발이 용이하지 않기 때문인 것으로 꼽히고 있다.

로봇 프로그램은 근본적으로 일반 컴퓨터 프로그래밍 시스템을 기반으로 설계되었기 때문에 기존의 컴퓨터 프로그래밍에 연관되어 있는 많은 어려운 문제들을 포함할 뿐만 아니라, 로봇 자체와 로봇이 상호 작용하고 있는 물리 세계와 연관되어 프로그램되어야 하기 때문에 어렵다. 오늘날의 산업용 로봇의 소프트웨어 대 하드웨어 비용은 비교적 간단한 로봇 작

업을 위한 싱글 로봇 어플리케이션의 개발이라도 로봇 자체 가격과 맞먹는 것이 일반적인 경향이다. 이러한 상황에 직면하여 로봇 프로그램을 쉽고 빠르게 생성할 수 있는 방법을 찾는 것은 당연하며, 로봇 프로그램이 쉬워질 수 있다면 제조시스템 또는 그 이외의 분야에서 로봇의 활용이 훨씬 빠른 속도로 증가할 것이다.

Task 수준 프로그램은 작업지시 수준의 로봇 프로그램으로 어렵고 난해한 로봇 프로그램을 로봇의 임무를 명시하는 수준에서 기술한다. 사용자가 목적하는 작업의 하부목표를 실행시키기 위하여 로봇이 하여야 할 모든 행동과 세부적이고 기계적인 데이터를 사용자가 직접 자세한 수준까지 기술하기보다는 Task 수준 프로그램 시스템에서 자동적으로 생성할 수 있어야 한다. 따라서 시스템에서는 작업자가 명시적 로봇 프로그래밍 언어를 기술했을 때보다 훨씬 더 고급수준의 지시를 로봇에 적용할 수 있는 것이다.

Shimon *et al.* (1993)은 Task를 이용하여 멀티 로봇 셀에서 로봇의 협동작업을 계획하는 CRP(Cooperation Requirement Planning) 방법론에 대해서 연구하였다. 그들은 CRP Process를 2 단계로 구분하였는데 1단계, 즉 CRP-I은 로봇과 Cell의 Assembly 제약 조건과 협동요구를 생성해 내는 것이며, 2단계, 즉 CRP-II는 Task를 할당하고 Task 수행을 위한 global plan을 조정하는 단계로 이루어져 있다. CRP는 조립작업시 조립구조를 파악하고 로

* 본 연구는 한국과학재단 지정 울산대학교 지역협력연구센터의 지원에 의한 것임.

봇의 협동작업을 Task로 기술하며 최적화된 global xxecution plan을 생성한다.

Albus(1993)는 RCS(A Reference Model Architecture For Intelligent Control; Albus, 1992) 구조를 갖는 시스템에서의 Task Decomposition을 위해서, Spatial Decomposition을 수행하는 Job Assignment Submodule, Agent의 계획을 생성하는 Planner집합, 그리고 Agent의 계획을 실행시키는 Executor 집합으로 구성되어있는 Task Decomposition Module에 대하여 연구하였으며, Task Decomposition에 필요한 Task Frame을 제시하고 Frame의 구성요소인 Task Name, Object, Parameter, Agent, Requirements, Procedures에 대하여 어떤 형식으로 정의되어야 하는지를 논하였다.

Haigh and Veloso(1997)는 고수준의 명령을 이용한 저수준의 명령어 생성에 관한 연구를 하였다. 이들은 ROGUE라는 User의 명령에 반응하여 경로를 설정하고 로봇의 Task Control Architecture로 명령을 전달하는 System Architecture를 개발하였으며, 동기적 또는 비동기적인 작업에 대한 하부작업을 생성하고 계획하는 Prodigy를 개발, 이용하였다. 작업의 목표를 달성하기 위하여 비동기적인 다중작업에 대한 계획방법, 작업수행의 감시와 실패를 예방하고 재계획하는 방법을 제시하였으며, 작업수행의 경험과 feedback을 이용한 학습을 통한 로봇의 행동의 개선을 시스템에 반영하였다.

Haigh and Veloso(1997)의 연구에 의하여 제시된 System Architecture는 Task 수준 프로그램을 계획하는 단계이며, 본 논문은 이 Task 수준 프로그램을 이용하여 로봇 제어를 가능하게 할 수 있도록 로봇 프로그램을 자동 생성할 수 있는 방법론에 대하여 다루고 있다. 따라서 본 연구는 앞서 수행된 연구의 결과와 더불어 로봇 시스템을 효과적으로 이용할 수 있는 기반 기술로 활용될 수 있을 것이다.

본 연구의 목적은 로봇을 이용하는 제조 cell에서 효율적이고 빠른 로봇 프로그래밍을 가능하게 하기 위하여 제조 cell의 작업 요구로부터 로봇의 작업요구, 즉 로봇의 Task 수준 명령을 생성하고, Task 수준 명령으로부터 로봇의 제어가 가능한 명시적 로봇 프로그램 (Explicit Robot Program, ERP)을 자동 생성할 수 있는 방법론을 제시하는 것이다. 또한, 이 방법론의 실용성을 검증하기 위하여 ERP 자동 생성 시스템의 프로토타입을 개발하였다.

2. ERP 생성방법론

로봇의 임무는 그 로봇이 위치한 작업 셀의 임무와 피가공품의 공정계획으로부터 유도될 수 있다. 예를 들어 어떤 작업 셀의 임무가 Part A를 가공하는 것이고 이 Part A의 가공을 위하여 공정계획은 기계 #1과 기계 #2에서 각각 가공되도록 계획되어 있다고 하자. 이때 그 작업 셀에서의 로봇의 임무는 다음과 같은 네 가지로 유도될 수 있다. 첫째 부품 A의 원자재를 입력 버퍼에서 집어 기계 #1에 적재하는 임무, 둘째 기계 #1에

서 중간 가공된 부품 A를 중간 버퍼로 이동하는 임무, 그리고 중간 버퍼에 있는 부품 A를 기계 #2로 적재하는 임무, 마지막으로 기계 #2의 부품 A를 출력 버퍼로 하역하는 임무 등으로 분류될 수 있다. 제조 셀의 작업임무를 완수하기 위해서는 이러한 로봇 임무들이 로봇에 전달되어 로봇이 이해할 수 있는 명시적 로봇 프로그래밍 언어로 변환되어야 한다.

본 논문에서 로봇 Task가 주어졌을 때 이를 로봇이 이해할 수 있는 명시적 로봇 프로그램(Explicit Robot Program)으로 자동 변환하기 위한 3단계 방법을 아래와 같이 제시한다.

Step 1(Task Decomposition): 로봇 Task를 로봇 수준 단위 Task (Decomposed Unit Task, DUT)로 분해한다.

Step 2(Task Synchronization): DUT를 Signal 및 위상정보와 함께 동기화 로봇 Task (Synchronized Unit Task, SUT)를 생성한다.

Step 3(ERP 생성): SUT를 명시적 로봇 언어(Explicit Robot Program, ERP)로 변환한다.

이와 같은 ERP자동 생성을 위한 3단계 방법이 <그림 1>에 표현되어 있다. Cell Task란 그 제조 셀에 주어진 생산임무를 말하며 생산 임무 각각에 공정계획을 적용하여 로봇 Task를 유추할 수 있는 원천 정보이다. 로봇 Task는 그 로봇이 속해 있는 셀의 목표를 달성하기 위하여 로봇이 해야 할 임무를 의미하므로 로봇이 수행할 행동을 세부적으로 명시하기보다는 사용자, 즉 셀의 목표 달성의 관점에서 기술된다. 예를 들어 "Part A를 기계 #1로부터 기계 #2로 이동하라"는 로봇 명령은 로봇이 해야 할 행동 내용을 상세히 기술하고 있지 않으며 부품 A를 가

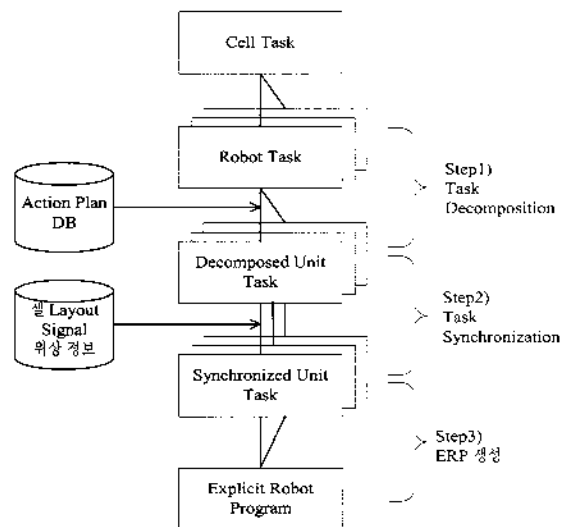


그림 1. ERP 자동생성을 위한 3단계 방법.

공하기 위한 목적의 일관임을 알 수 있다.

일련의 로봇단위업무들이 순서에 입각하여 이루어짐으로써 하나의 로봇업무가 이루어지며, 이때 단위업무들은 순차적으로 발생하는 경우, 2개 이상의 일이 동시에 병렬적으로 발생하는 경우, 그리고 어떠한 행동이 발생해야 비로소 다음 행동이 발생할 수 있는 동기적인 경우로 나누어 질 수 있다. Action Plan이란 하나의 로봇 임무가 이루어지는 세부 내용을 표현하는 그래프로 로봇 업무의 순서제약과 단위 업무들의 표현을 포함한다. Action Plan은 로봇 임무가 이루어지기 위하여 꼭 필요한 단위업무들로 구성되어 있으며, 특정 제조시스템과 독립적으로 표현된다. Petri-net은 Action Plan의 순서 제약을 모두 표현할 수 있는 그래픽 도구이므로 Action Plan을 표현하는 데 이용된다. <그림 2>는 Petri-net을 이용하여 표현된 로봇의 "Pick Task"의 Action Plan을 나타내고 있으며 로봇이 어떠한 사물을 놓을 때 나타나는 행동을 기술하고 있다. Pick Task는 첫째 현재위치를 확인하고(P1), 목표상태로 가기 위해서 장애물이나 진입경로상의 위치(P2)를 거쳐서 목표상태(P3)에 도달하고, Gripper를 Open(P4)한 다음, 다음 명령을 수행할 수 있는 안전한 위치(P5)로 이동하는 일련의 단위업무(DUT)들을 통하여 이루어지고 있음을 표현하고 있다.

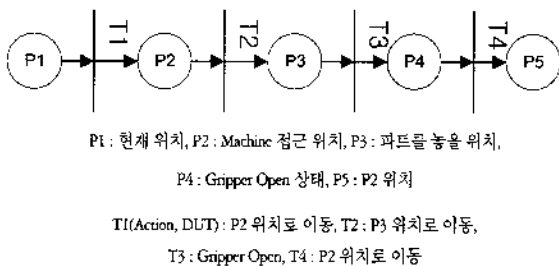


그림 2. 로봇 Pick Task의 Action Plan.

로봇단위업무(Robot Unit Task : RUT)란 일단 시작되면 끝날 때까지 외부 환경과 상호작용 없이 수행되는 성격을 갖고 있다. RUT는 로봇업무가 분할된 Decomposed Unit Task(DUT)와 각 DUT들이 외부와 상호 의사소통하기 위하여 요구되는 단위 업무인 Synchronized Unit Task(SUT)로 구분된다. DUT는 분해된 로봇의 물리적인 행동을 기술하는 동작기술을 위한 것으로 Action Plan에 제조시스템의 종속적인 데이터, 즉 Layout의 위치를 나타낼 수 있는 위치정보와 장비의 진입 경로 정보, 장비의 다른 독특한 작동양식 그리고, Part에 대한 형상정보 등의 종속적인 정보가 포함된 것이다. SUT는 제조시스템 내의 셀 controller 또는 기계장비와 통신을 위한 동기화 정보가 포함된 것이다.

예를 들면 "Gripper를 닫아라/열어라"는 동작기술 RUT, 즉 DUT이며, 부품 A를 기계 #1에 놓을 때 기계 #1에 부품 A를 놓아도 되는지를 알기 위해 Signal 교환을 수행하는 단위업무가

포함된 동기화 RUT, 즉 SUT이다.

Explicit Robot Program은 로봇이 취하는 모든 동작이 프로그램된 것으로, 동기화 로봇단위 Task와 물리적 환경정보를 이용해 로봇을 직접 실행시킬 수 있는 로봇 프로그램이다. 명시적 프로그램은 로봇의 매니플레이터 프로그램을 위한 특수한 기능을 갖는 것이 보통이므로 흔히 로봇 프로그래밍 언어(Robot Programming Languages: RPLs)라고 할 때는 이 ERP를 의미하는 것이다.

2.1 로봇 임무의 분해

로봇 Task로부터 DUT를 생성하기 위해서는 부품과 구성장비의 정보를 기초로 하여 로봇의 단위 이동경로와 로봇 Task로부터의 정보, Part를 쫓 것인지 아니면 놓을 것인지에 따라서 결정된다. 아래 <그림 3>은 로봇의 Task를 DUT로 생성하는 것을 표현하는 것으로 현실의 Layout과 관련한 Part 및 구성장비가 포함하는 특성정보(본 연구에서는 로봇의 접근경로만을 고려하였다.)를 Task Type에 따른 Action Plan의 기본적인 행동양식을 기준으로 하여 그에 상응하는 Instance를 생성하는 것이다.

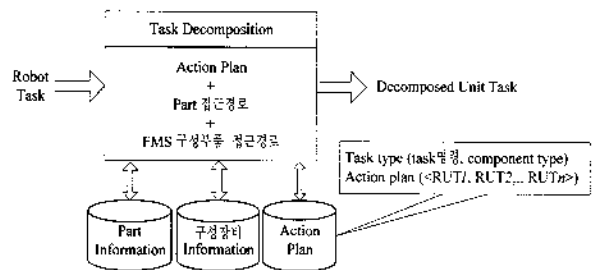


그림 3. 로봇 임무 분해.

로봇 Task의 분해단계를 살펴보면 다음과 같다.

- Step 1: 로봇 TASK의 타입을 분류하여 ACTION PLAN 획득
- Step 2: 대상이 되는 PART 정보로부터 PART 접근경로 추출
- Step 3: 대상이 되는 COMPONENT 정보로부터 COMPONENT 접근경로 추출.
- Step 4: ACTION PLAN에 PART 접근경로와 COMPONENT 접근경로를 적용시켜 DECOMPOSED UNIT TASK 생성

로봇의 Task Type은 로봇이 어디에서 어떠한 작업을 수행하는가를 중심으로 분류한다. 예를 들면 Input-Pick, Machine-Place, Machine-Pick, Output-Place, Buffer-Pick, Buffer-Place 등이 있으며,

이에 따른 각각의 Action Plan을 획득할 수 있다. Part 접근경로는 로봇이 파트를 쥐려고 할 때 파트에 접근하기 위한 경로이며, 연속적인 좌표계로 표현한다. 본 논문에서 각각의 파트에 접근의 기준이 되는 좌표계를 파트의 원점 좌표계를 기준으로 생성해주었으며, 이때 생성된 일련의 좌표계를 접근경로라 한다. Component의 접근경로도 파트 접근경로와 마찬가지로 좌표계를 생성하여 파트 또는 로봇이 기계와 충돌이 생기지 않도록 미리 정의해 놓은 좌표계를 통해서 파트가 놓일 장소, 또는 놓여 있는 장소로 이동할 수 있도록 하는 접근경로이다. 이러한 로봇의 접근경로는 로봇의 Task 타입에 따라서 로봇이 파트를 포함한 상태에서 움직이는 것과 아무 것도 없는 상태에서 움직이는 두 경우를 고려하여 DUT를 생성하게 된다.

2.2 임무 동기화

로봇의 Task가 올바르게 수행되기 위해서는 제조 셀을 구성하는 다른 구성장비들과 정보를 교환하여야 한다. 로봇 단위 Task와 작업 셀의 동기화(Synchronized Robot Task)를 통해 로봇에 주어진 임무를 수행할 수 있다. 동기화 로봇 Task는 로봇과 Controller, 로봇과 Component의 Sensor 사이의 통신관계를 Component의 Type과 로봇 Task의 종류에 따라서 구분할 수 있다. <그림 4>는 DUT를 다른 구성요소와 함께 상호작용을 할 수 있게 하기 위한 SUT를 생성하는 것으로 각각의 DUT를 그 Type에 따라 분류하여 해당 RUT 전후에 동기화에 관련된 RUT를 삽입하는 것이다.

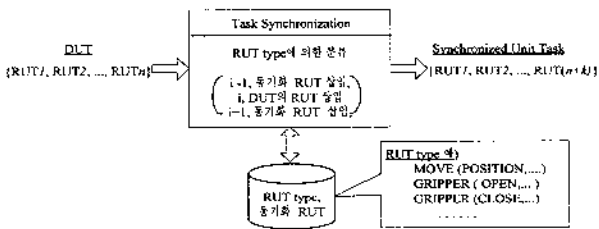


그림 4. Robot Unit Task의 동기화.

각 Component 타입과 로봇 Task에 따른 통신관계를 명확히 정의하기 위하여 Petri-net을 이용해서 어느 부분에서 동기화가 발생할 것인지를 예측하여 이를 SUT 생성에 적용하였다. 이를 위해서 구성품 타입과 로봇 Task의 종류에 따라서 Petri-net의 특징인 순차적 발생, 동시발생, 동기화를 DUT에 첨가하여 SUT를 생성하였다. 순차적 발생이란 단위 Task의 시간적인 순서를 나타내는 것이며, 동시발생이란 Sensor, 또는 Controller로 가는 신호송신 상태를 표현한 것이다. 그리고 동기화란 Controller 또는 구성요소의 Sensor와 Sub-Task 간의 동기화를 표현한 것이며 이는 Signal 수신에 해당된다.

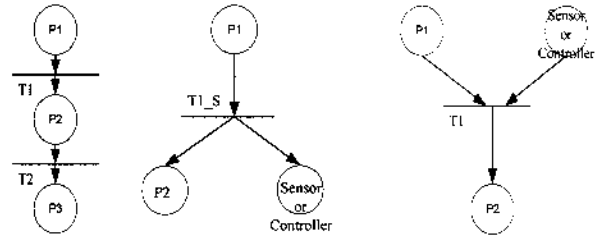


그림 5. DUT를 SUT로 만들어 주는 Petri-net의 특징.

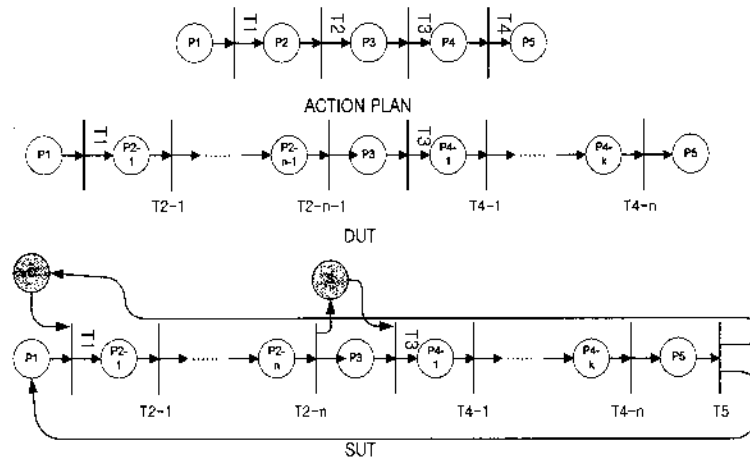
동기화 로봇 단위 Task를 유도하는 절차는 다음과 같다.

- Step 1: DUT의 모든 RUT에 대하여 RUT TYPE을 분류한다.
- Step 2: 분류된 RUT TYPE에 따라서 동기화 RUT들을 생성한다.
- Step 3: 동기화 규칙에 따라 생성된 RUT를 SYNCHRONIZED UNIT TASK에 삽입한다.
- Step 4: DECOMPOSED UNIT TASK에서 이용된 RUT를 제거하고 DECOMPOSED UNIT TASK가 존재하지 않을 때까지 반복한다.

동기화 규칙이라 함은 <그림 5>와 같이 미리 정의되어 있는 순차적 실행, 동시발생, 동기화에 관련된 데이터를 <그림 2>와 같이 각각의 RUT Type과 Component의 Type에 따라 삽입 위치를 규칙으로 정의해 놓은 것이다. 예를 들면 Gripper를 Open하기 위해서 그 RUT에 관련된 구성요소에게 Gripper Open 허가 메시지를 전달하는 것을 규칙으로 표현한 것이다. 이러한 규칙들의 집합을 동기화 규칙이라 한다. <그림 6>은 Input Type의 Place작업을 수행하는 로봇의 행동을 표현하는 DUT와, Controller와 Component, Controller와 Sensor의 동기화를 표현한 SUT이다.

2.3 Explicit Robot Program (KAREL) 생성

명시적 프로그램은 Work Cell의 물리적 자료, 즉 Layout, Component, 그리고 Part의 물리적 정보를 이용하여 실질적으로 로봇을 구동할 수 있도록 해준다. 본 논문에서는 명시적 프로그램언어로 KAREL을 선택하였으며 KAREL 프로그램은 로봇이 움직여야 할 위치와 자세의 정보를 포함하는 Position부분과 프로그램의 흐름을 나타내는 Procedure부분으로 나누어져 있다. <그림 7>은 SUT를 로봇에서 실행할 수 있게 하기 위한 변환을 표현한 것이다.



- P1 : 현재 위치, P2 : Machine 접근 위치, P3 : 파트를 놓을 위치, P4 : Gripper Open 상태, P5 : P2 위치
- P2-1 ~ P2-n-1 : Machine 접근 위치부터 P3 직전 위치, P2-n : P3 위치
- P2-1 ~ P2-k : Machine Gripper Open 상태에서부터 P5 직전 위치
- T1(Action, DUT) : P2 위치로 이동, T2 : P3 위치로 이동, T3 : Gripper Open, T4 : P2 위치로 이동
- T2-1 ~ T2-n-1 : P2-n 위치까지 이동, T2-2 ~ T2-k : P5 위치까지 이동
- T1(SUT) : Controller로부터 메시지를 받고 현재 이동 가능한 상태이면 P2-1 위치로 이동
- T2-n : Sensor로 Gripper Open 허가 요청
- T3(SUT) : Sensor로부터 Gripper Open 허가 메시지를 받고, P3 위치에 있으면 Gripper를 Open
- T4(SUT) : Controller로 Task가 끝났다는 메시지를 전송하고 대기상태로 전환

그림 6. Input타입의 Component와 Pick Task에 대한 SUT.

Synchronized Unit Task

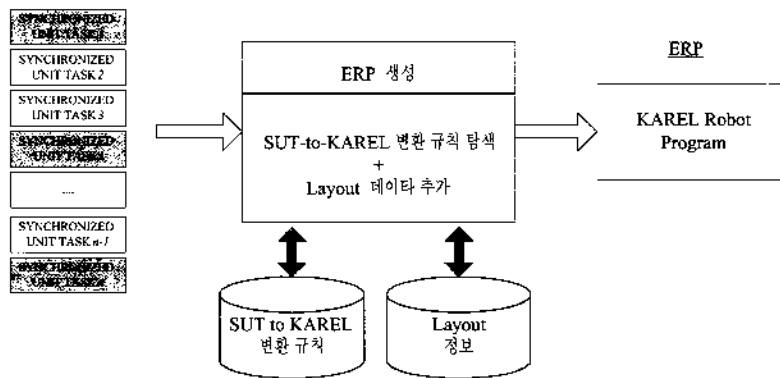


그림 7. Explicit Robot Program (KAREL) 생성.

로봇의 자세를 구하기 위해서 KAREL에서는 로봇의 작동에 필요한 파라미터를 요구한다. 요구되는 파라미터들은 Tool Point와 단말장치의 Yaw, Row, Pitch이다. 이 같은 파라미터들은 실세계를 기준으로 한 Layout 좌표계와, Part와 Component의 좌표계를 기준으로 Part와 Component의 접근 좌표계들로부터 로봇의 자세를 정하기 위해서 로봇의 좌표계를 기준으로 Transformation함으로써 구하게 된다. KAREL의 Procedure 부분은 SUT를

KAREL 프로그래밍언어로 변환시켜 생성하게 된다. 본 연구에서는 SUT와 ERP 사이의 변환을 용이하게 하기 위하여 SUT와 KAREL명령어 사이에 <표 1>과 같은 변환관계를 갖게 하였다. 본 연구에서 이용된 RUT인 MOVE RUT는 KAREL에서의 MOVE POSITION의 명령어로 GRIPPER RUT는 KAREL의 GRIPPER 관련 명령과 일치하게 하였다. 또한 상호관계를 표현하는 RUT인 Signal SEND, RECEIVE RUT는 KAREL의 DIN,

표 1. 단위 Task와 KAREL 변환 규칙

Petri-net의 Transition 종류	자료 표현	KAREL
이동에 관련된 Transition	MOVE(..Position..)	MOVE TO POSITION
GRIPPER에 관련된 Transition	Gripper(..Open..)	Open hand Robot
	Gripper(..Close..)	Close hand Robot
Signal 송수신에 관련된 Transition	Signal(..Send.Signal..)	--! SIGNALDEF DOUT[Signal], NAME, Signal DOUT[Signal]=ON
	Signal(..Receive.Signal..)	--! SIGNALDEF DIN[Signal], NAME, Signal WAIT FOR DIN[SIGNAL]=ON

DOUT, WAIT를 이용하여 변환되게 하였다. 동기화 로봇단위 Task의 KAREL 변환절차는 다음과 같다.

Step 1: SUT의 모든 RUT에 대하여 변환규칙을 탐색한다.

Step 2: MATCH된 변환규칙에 따라서 KAREL 로봇 PROGRAM을 생성한다.

Step 3: SUT에서 이용된 RUT를 제거하고 SUT가 존재하지 않을 때까지 반복한다.

Step 4: 반복이 끝나면 생성된 KAREL 로봇 프로그램에서 필요한 수치 DATA를 생성하고 하나의 PROCEDURE로 변환시킨다.

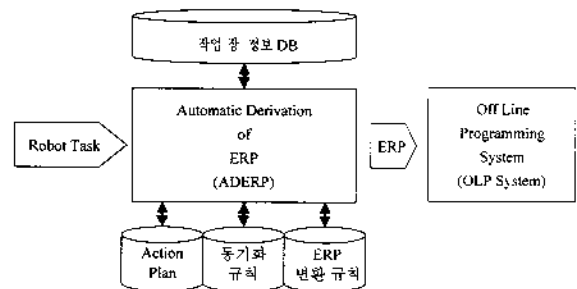


그림 8. System 구조

3. 구현

본 논문에서 제안된 3단계 ERP 생성방법론을 검증하기 위하여 ERP 자동생성을 위한 프로토타입 시스템을 개발하였다. 본 시스템의 구조는 로봇 Task로부터 ERP 프로그램을 생성하는 ADERP (Automatic Derivation of ERP) 시스템과 이 시스템으로부터 생성된 ERP 프로그램의 타당성을 검증하기 위한 OLP 시스템으로 이루어져 있으며 <그림 8>에 그 System 구조를 표현하였다.

본 프로토타입에서 이용된 Work Cell은 Cell 내에 input, output, machine1이 존재하며 Part A를 가공하는 Cell이다. Part, Component의 정보와 Layout정보는 임의로 정하여 시스템에 적용해 보았으며 Pick (Robot1, Machine1, Part1)이라는 명령을 <그림 9> (a)와 같이 내렸을 때 <그림 9> (b)와 같은 출력을 얻을 수 있었으며 생성된 로봇 프로그램을 OLP시스템을 통해서 <그림 9> (c)와 같이 검증하였다. Off Line 프로그래밍 시스템

(OLP)은 컴퓨터 그래픽스의 방법에 의하여 충분히 확장된 로봇 프로그램 개발 환경으로 로봇에 직접 접근하지 않고도 로봇 프로그램을 개발할 수 있는 시스템이다.

4. 결론 및 향후 연구과제

본 시스템은 로봇의 다기능성 중 무인자동화를 위한 로봇 중심의 Work Cell 관점에서 로봇 프로그램 자동생성 시스템을 제안하였으며, Task 단위의 제어가 가능한 로봇 프로그램을 자동으로 생성하는 시스템의 프로토타입을 개발하였다. 본 연구에서 제안한 프로토타입 시스템은 Work Cell의 설계시 또는 기존 Work Cell 변경시의 Layout, 생산품 그리고 생산방법에 따른 변화를 신속하게 반영하여 대처할 수 있으며, 이러한 시스템은 제품 생산주기가 짧아지면서 발생하는 Layout을 비롯한 여러 가지 환경의 잦은 변화에 신속히 대처할 수 있으며 Work Cell의

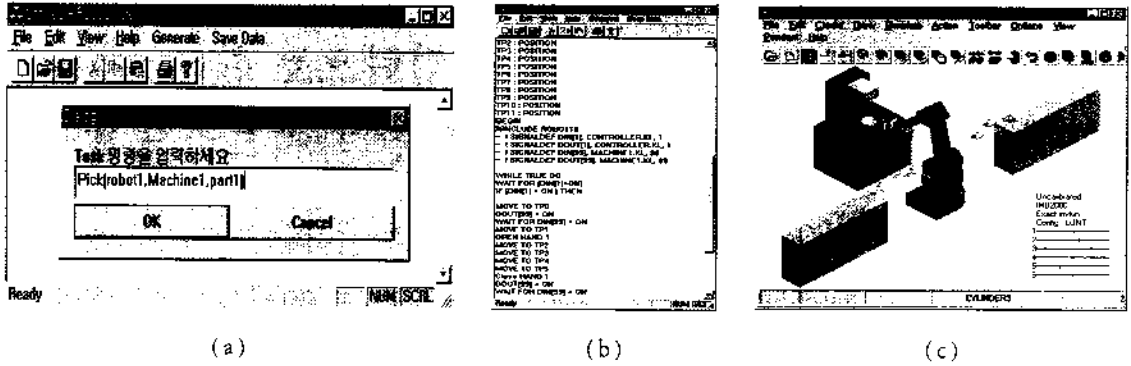


그림 9. (a) Task 명령, (b) 출력, (c) 검증.

유연성을 최대화시켜줌으로써 제조공장의 무인화를 한층 앞당길 수 있을 것이다. 그리고 본 프로토타입의 시스템의 결과를 살펴보면 평균적으로 Task를 입력받아서 로봇 프로그램을 생성하는 데 1초(Pentium 166MHz에서 실험) 이내의 성능을 보여 주었다. 즉, 로봇으로 Uploading 방법만 개선시킬 수 있다면 로봇의 실시간 제어가 가능함을 보여준다. 향후 연구과제로서는 Task 수준의 로봇 프로그램을 효율적으로 이용할 수 있는 적절한 Work Cell Controller 개발이 필요하며, 효율적인 로봇 경로 유도과 UNIT TASK와 작업 구성요소들 간의 동기화에 대한 여러 가지 측면에서 연구해야 할 것이며, 실시간 제어분야에도 적용해 보아야 할 것이다.

참고문헌

James, S. A. (1992), RCS: A reference model architecture for intelligent control, *IEEE Journal on Computer Architectures for Intelligent Machines*, May.
 James, S. A. (1993), Task decomposition, *Proceedings of the 1993 International Symposium on Intelligent Control*, Chicago, IL, Aug.
 John, J. C. (1992), *Introduction to Robotics Mechanics and Control*, Addison-Wesley Publishing Company Inc.
 Karen, Z. H. and Manuela, M. V. (1997), High-level planning and low level execution: Towards a complete robotic agent proceedings of the first, *International Conference on Autonomous Agents*, Feb.
 Shimon, Y. N., Venkat, N. R. and Purdue (1993), Automatic generation of assembly constraints and cooperation task planning, *Annals of the CIRP*, 42(1).



서원호
 고려대학교 산업공학 학사
 펜실베이니아 주립대학교 석사, 박사
 현재: 울산대학교 수송시스템 공학부 교수
 관심 분야: CAPP, 사례기반 설계, 로봇 운용,
 가상제조를 이용한 제조시스템 설계



정덕호
 울산대학교 산업공학 학사
 현재: 울산대학교 산업공학과 석사과정
 관심 분야: 제조시스템 자동설계, 사례기반
 추론