論文 한국정밀공학회지 제14권 제2호 (1997년 2월)
Journal of the Korean Society of Precision Engineering Vol. 14. No. 2. FEBRUARY. 1997.

UniSet개발을 통한 공장자동화에 관한 연구

최 경 현*, 김 성 청**

A Study of the Automation of Factory through the Development of UniSet

Kyung-Hyun Choi*, Sung-Chung Kim**

ABSTRACT

This paper reports the effort for developing this new Unified Manufacturing Instruction Set and its environment, called here UniSet, to deal with difficulties in set up and operation of Flexible Manufacturing Cells. UniSet has been developed as a non-exclusive unified manufacturing instruction set based on comparisons of the prevailing machine tool and programming primitives. UniSet allows programmers to deal with only one instruction set, if they so desire, in a single coherent environment, rather than numerous machine programming languages. The software system is coded in an Object-Oriented Programming (OOP) language, Smalltalk, and derives its paradigm from the OO philosophy. Test results are also included to demonstrate the applicability of the approach employed.

Key Words: Flexible Manufacturing Cell(유연제조셀), Cell Programming and Control(셀프로그래밍과 제어), Object-Oriented Programming(객체지향프로그래밍), CIM(컴퓨터 통합제조)

1. 서 론

공장 자동화 연구에 있어서 가장 최근의 발전 가운데 하나가 유연제조시스템(Flexible Manufacturing Systems)이다. 이상적인 유연제조시스템에서는 최고 경영에서부터 공장 현장까지(shop floor)의 모든 서류 교환이 사라지고 생산활동에 관련된 모든 정보의 교환은 컴퓨터와 기계 제어기 사이에서 자동적으로 이루어진다. 유연제조시스템은 자재 운반장치(AGV나 혹은 컨베여 등)로 연결된 유연제조셀(Flexible Manufacturing Cells)을 기본 단위 체로 하여 구성 될 수 있다. 유연제조셀은 일반적으로 중소기업에 있어서 유연성과 생산성을 향상시킬 수 있는 최상의 도구로 여겨지고 있으며, 기본 구성 요소로는 다양한 부품들을 자동 가공하거나 부품들을 유니트에 조립하는데 사용되어지는 NC 공작기계 혹은 워크스테이션, 로봇과 같은 자재 취급 장치 그리고 이들을 통합 제어

* 한국원자력 연구소
** 충북대학교 기계공학부

에 사용되는 제어 컴퓨터 등으로 구성되어 있다.

일반적으로 이들 구성 기계들은 각각 다른 벤더(multi-vendor)에서 공급되는데, 이들 벤더들은 고유 기계의 성능 향상에 초점을 맞춘 제어 언어의 개발에 노력을 하고 있는 반면, 다른 벤더에서 제작되는 기계들과의 통합을 위한 제어시스템의 설계는 고려하지 않는 실정이다. 이러한 멀티-벤더 기계들을 조합하여 유연제조셀을 실제 구성하고 운용하는데는 여러 가지 장애요소들이 존재하고 있다. 이 중 가장 큰 장애요인으로는 구성 기계들이 실행하고 있는 프로그래밍 언어 및 환경 그리고 커뮤니케이션 프로토콜의 상이성(incompatibility)으로 분석되었다. 이러한 생산환경이 제조셀을 효율적으로 구축하고 운영하기 위해서는 보다 다양해지고 생산량의 로트 단위가 줄어드는 실정에서 로봇이나 NC 공작기계 등의 작업 프로그램을 작업자가 해당기계언어로 작성해야 함으로써, 각 기계들의 특성을 이해하고 취급하는 전문가가 필요하며, 셀 전체의 프로그램 시간이 대단한 장애요인으로 등장하게 된다. 따라서 공장기계가 다양해질수록 고숙련 노동자가 요구되고 setup 및 생산 시간과 복잡도가 증가한다. 결과로 자동화에 따른 생산비용이 증가하며, 이것은 자동화 취지에 상충되는 것이다.

이러한 문제를 다루기 위해 북미회사인 Kearney & Trecker 및 Cincinnati Milacron은 자사 기계로만 구성된 유연제조셀들을 개발했으나, 유연성의 부족과 높은 비용으로 인해 구매자들의 관심을 유도하지 못해 실용성에 의문이 제기되었다. 더욱이 유연제조셀을 구성하는 모든 기계들을 하나의 벤더에게만 의존하는 방식은 다른 벤더의 경쟁력이나 혁신기술을 이용할 수 없어 비생산적인 것으로 밝혀졌다. 대안적으로 제시된 연구 경향은 일반적이고 표준적인 프로그래밍 언어나 커뮤니케이션 프로토콜을 개발하는데 집중하거나, 혹은 로봇언어인 AML을 주로 사용한 Grossman[1,2], 및 Sharon[3]의 연구에서 볼 수 있듯이 제조셀에서 멀티-벤더 기계 문제를 다루기 위해 기존의 로봇이나 생산 언어(Manufacturing Language)를 확장하는데 초점을 맞추었다. 이 경우 벤더들은 이러한 언어의 요구에 시스템을 변경하거나 포스트 프로세서를 개발하여 공급 해야하는데, 일반적으로 벤더들은 기존의 커뮤니케이션 프로토콜이나 혹은 제어 언어를 선호하고 있다.

최근의 연구는 셀의 구성 기계들은 해당기계언어로 프로그램 할 수 있도록 하고, 오직 제조셀을 위한 제어구조와 언어 개발에 집중되어 있다(Li[4], Magy외 [5]). 이러한 연구들은 제어 소프트웨어를 개발하는데 객체 지향적인 접근법을 사용함으로써 제조셀내의 빈번한 변화를 다루는 유연성이나 용이성을 가능케 하려고 노력하였다. 그러나 아직도 셀 프로그램에 있어서는 다양한 해당 기계 언어가 사용되어야 한다.

이 논문에서 유연제조셀의 멀티-벤더 기계 문제를 다루는데 적용한 접근법은 기존의 연구에서 제안한 접근법과는 다른 방법론을 제안하고 있다. 유연제조셀의 구성 기계들이 사용하고 있는 자동 생산언어(Automatic Manufacturing Language)의 총체적 연구와 유연제조셀을 setup하고 재조정(reconfigure)할 때 부딪치는 문제들을 분석한 결과로 UniSet(Unified Instruction Set)을 개발하였고, 이를 통한 셀 전체의 운영 프로그램을 단일 언어를 사용하여 구현할 수 있는 이점을 부여한다. 개발된 UniSet의 개념 및 개발의 방법론이 구체적으로 언급될 것이다.

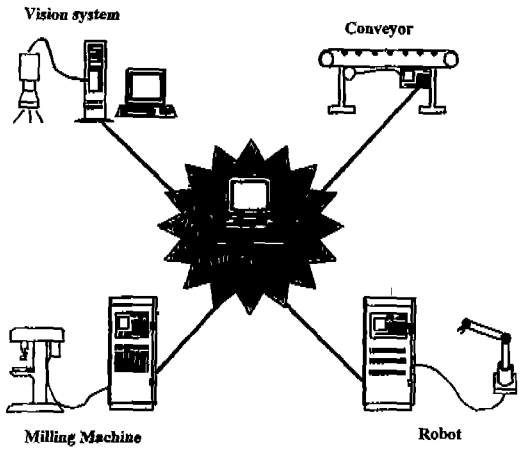

Fig. 1 Typical cell components and communication in an FMC

2. UniSet의 개념

NC 공작기계, 로봇, 컨베어, 센서시스템 등이 유연제조셀을 구성하는 주요 장비들이며, 셀 제어기의 명령에 의해서 주어진 작업을 수행하기 위해 그림 1에서 보여지는 것처럼 셀 제어 컴퓨터와 구성 기계들이 스타 네트워망 구조로 연결되어야 한다. 이러한 구조는 셀 제어 컴퓨터의 확장에 의해 시스템 전체의 확장성이 뛰어나며, 한 개의 기계가 고장나더라도 전 시스템에 미치는 영향이 아

주 적으며, 또한 셀 제어 프로그램을 바꿈으로서 시스템 제어를 간단히 바꿀 수 있어 유연성을 높일 수 있는 장점을 부여하고 있다.

그림 2에서 처럼 UniSet은 포괄적인(generic) 셀 프로그래밍 환경을 공급하며, 제조셀 작업자가 구성기계 만큼 다양한 해당기계 언어보다는 전체 셀을 단일한 명령어를 사용하여 프로그램을 할 수 있도록 하였다. 셀 프로그래머가 머시닝 센터를 위한 NC 파트 프로그램을 작성할 때, 구성 기계요소들, 즉 spindle, tool 등의 상호 작용을 서술 하듯이 유연제조셀을 구성하는 기계들을 다양한 방법으로 상호 작용하는 요소로 여겨 UniSet으로 셀 구성 기계들이 원자재를 어떻게 취급하고 가공하는가를 서술하면 된다. 번역 모듈을 개발하여 UniSet 명령어들을 해당 기계 언어의 명령어 형태로 변환시킨후 해당기계로 다운로드하여 실행한다. 이러한 접근법은 평범한 작업자가 셀 프로그램을 수행 할 수 있을 뿐만 아니라 프로그래밍 시간을 줄일 수 있는 장점을 부여한다.

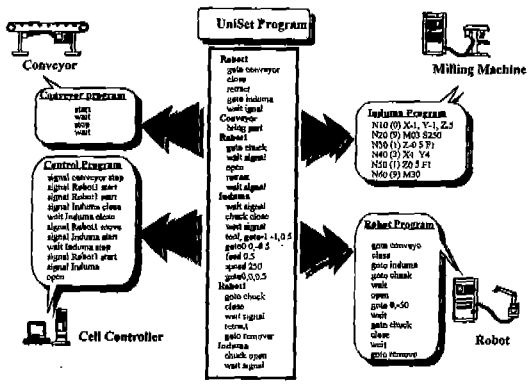

Fig. 2. UniSet Concept and Program Translation

UniSet의 다른 주요 장점은 셀 프로그래밍에 있어서 배타적인 접근법을 사용하지 않음으로서 새로운 기술을 수용할 수 있다. 예를 들면 NC 공작 기계의 공구 경로를 위한 CAD/CAM에 의해 생성된 프로그램들을 직접 포함할 수 있다.

3. UniSet의 개발

본 연구에서는 제조셀을 구성하는 NC 공작 기계 및 로봇이 실행하고 있는 프로그래밍 언어들의 명령어 사이에는 밀접한 관계가 있음이 분석되었다. 각 명령어들의 기능은 같지만 포맷은 다르게 각 언어에서 정의되었다. 예를 들면, 한 지점에서 다른 지점으로 옮기는데 VAL II에서는 MOVE를 APT에서는 GOTO라는 명령어를 사용한다.

UniSet의 개발은 현재 널리 사용되고 있는 자동생산언어(Automatic Manufacturing Languages)인 VAL II, Word Address, ASEA, APT들의 명령어들을 비교분석 후, 그들 명령어들의 기능을 모두 포함하고 셀을 구성하는 모든 기계들을 관리/제어 할 수 있는 명령어까지 포함하는 실질적인 자연언어에 가까운 명령어 포맷을 갖도록 설계 되어졌다.

개발된 UniSet 명령어는 해당 기계들에서 같은 기능들을 수행할 수 있도록 객체지향언어인 Smalltalk에서의 다중성(polymorphism)의 성질을 이용한 메소드를 포함하는 객체들로 정의되었다.

3.1 UniSet 명령어 포맷

개발된 UniSet 명령어들은 표 1에서 보는바와 같이 여섯 가지 포맷으로 나누어진다. 첫 번째 명령어 포맷은 한 지시어로 이루어진 UnaryCommand로서, 변수 없이 반복되는 기능들을 정의하는데 사용되는 명령어 형태이다. 예를 들면, 로봇의 gripper를 열기 위해 변수 없이 Open 명령어를 쓴다. 두 번째 명령어 포맷은 변수를 가지는 명령어로서 지시어와 변수는 콜론(:)으로 구분되는 Command이다. 세 번째 명령어 포맷은 기초지시어와 이 지시어를 제한하는 한정어로 이루어진 ModifiedUnaryCommand인데 기초지시어는 APT에서의 기초 단어(major word)와 같은 성격을 지닌다. 네 번째 명령어 포맷은 변수를 갖는 한정어와 지시어로 구성된 UnaryModifterCommand이다. KeySymbol은 기하학적 묘사를 정의하는데 사용될 다섯 번째 포맷이다. 마지막 명령어 포맷은 BinaryModifierCommand로서 UnaryModifierCommand에 한정어가 추가된 형태이다.

UniSet 명령어에서 사용되는 심볼들 중 콤마(,)는 지시어와 한정어, 지시어와 지시어를 분리하는 역할을 하며, 중괄호{}는 적어도 한번은 일어난다는 표현이고, 대괄호()는 선택적 표현이고 수직바(|)는 선택적인 용어들을 분리하며 괄호()는 선택용어들의 집단을 표시한다.

Table 1. UniSet Instruction Format

Class	Format	Example
UnaryCommand	command	stop
Command	command <arg>	go:(1@2@3)
ModifiedUnaryCommand	command, modifier	Mist, off
UnaryModifiedCommand	command, modifier:<arg>	go,rapid:(1@2@3)
KeySymbol	command, <symbol>:<arg>	point, P1: (1@2@3)
BinaryModifiedCommand	command, modifier, modifier:<arg>	go,left,tool:(1@2@3)

3.2 명령어 합성

자동생산언어 명령어들의 비교와 UniSet 명령어를 정의 과정을 설명하기 위해 움직임을 나타내는데 사용되는 두 부류의 명령어를 예로서 사용하였다. 이들 움직임을 정의하는 명령어들은 로봇의 엔드 에펙터나 NC 공작기계의 공구 움직임을 서술하기 위해 사용되는 명령어로서 NC 공작기계나 로봇언어에서 가장 기초적이고 복잡한 명령어들이다. 이들 명령어들은 세 가지 기본 움직임을 정의하기 위해 조인트 운동, 직선운동, 회전운동의 세 가지 범주로 구분되고, 첫 두 가지 동작에서는 로봇이나 NC 공작기계의 움직임을 절대모드와 상대모드에 따라 서술될 수 있다. 로봇과 NC 공작기계 공구의 위치들이 절대 모드에서는 고정된 좌표계로 부터 정의될 수 있고, 상대 모드에서는 현 위치는 전 지점으로부터 상대적으로 나타낼 수 있다.

3.2.1 조인트 보간 운동 명령어

여기서 조인트 보간 운동은 다른 축 드라이버들의 운동이 조정되지 않은 결과로 얻은 엔드 에펙터나 공구의 움직임을 정의하는 일반적 용어로 사용된다. NC 공작기계에서는 이러한 움직임을 "point-to-point"를 의미한다. 이것은 주로 정확한 경로가 중요하지 않을 경우의 빠른 움직임에 이용된다. 예를 들면, 로봇의 pick/load 공정이나 좀 더 정확한 움직임을 위한 준비단계로서 공구를 위치시키는 경우에 사용된다.

자동생산언어에 있어서 절대 모드와 상대 모드에 대하여 조인트 보간 운동 명령어의 비교와 상응하는 UniSet 명령어의 정의가 표 2와 3에 보여진다. 언어 Word Address의 조인트 보간 운동은 G00코드와 목표 위치의 x, y, z 좌표 및 NC공작기계의 3축이상인 경우 a와 b로서 기술되며, 코드 G90과 G91은 각각 절대운동과 상대운동을 정의하기 위해 사용된다. 가공을 위해 공구를 위치시킬 때, 고려해야 될 중요 사항은 cutter 지름 보정인데 기능은 전형적으로 NC 제어기에서 수행되어 진다. 코드 G40, G41, G42는 각각 보정 취소(compensation cancel), 왼쪽보정(compensation left), 오른쪽 보정(compensation right)을 위한 명령어이며, XY평면(코드 G17), XY평면(코드 G18), XY평면(코드 G19)에서 일어날 수 있다. 이러한 명령들은 사용자가 정확한 공구 중심 위치를 계산하는 것을 생략할 수 있게 해준다. 이러한 보정은 절대 운동 모드에서만 유효하다.

APT에서 운동방식은 각각 절대모드와 상대모드 모두 명령어 GOTO와 GODLTA전에 RAPID를 기술함으로써, 조인트 보간 운동을 정의한다. 명령어 GOTO 또는 GODLAT가 이송률과 함께 기술될 때 조인트 보간 운동 중 contouring 운동으로 정의된다. APT가 주로 NC 공작기계를 위한 것이기 때문에, 이 접근법은 위치 결정과 가공을 위한 공구 운동을 지시하는데 매우 효율적인 방법이다. GOTO에 필요한 변수는 미리 정의된 위치의 심벌이나 x, y, z, 좌표로 정의되며, 선택적인 벡터 *i*, *j*, *k*는 3개 이상의 축을 가진 NC 공작기계들을 위한 공구의 방향을 정의한다.

ROBOT 언어인, VAL II는 pick/place 공정들을 위한 특별히 만들어진 네게의 명령어들을 제공한다. 절대 운동 방식을 위해서 MOVE와 MOVET명령어가 정의되었는데, MOVE는 로봇 엔드 에펙터에게 지정된 위치와 방향으로 움직이는데 사용되고, MOVET는 운동하는 동안 엔드 에펙터의 열림의 폭을 변화시킨다는 점에서 MOVE와 다르다. 상대 운동 모드를 위해서 APPRO는 로봇 엔드 에펙터에게 현위치와 방향에 대해 상대적으로 기술된 위치와 방향으로 혹은 로봇 Z축을 따라 오프셋의 값에 의해 조정된 위치로 움직이는데 사용되는 명령어이며, DEPART는 엔드 에펙터에게 오프셋에 명시된 거리로 로봇 Z축을 따라 움직이는데 필요한 명령어이다.

ASEA 로봇에서는, 조인트 보간 운동을 위한 명령어 POS전에 ROBOT COORD가 기술되어야한다. 명령어 POS은 위치를 정의하기 위해 x, y, z 좌표와 손목좌표계

Table 2. Joint Interpolation; Absolute Mode

<table>
<tr><td rowspan="7">Word Address</td><td colspan="3">point to point</td><td>G90[G40]G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td rowspan="6">Tool compensated motion</td><td rowspan="2">XY plane</td><td>LF</td><td>G17G41G90G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td>RT</td><td>G17G42G90G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td rowspan="2">XZ plane</td><td>LF</td><td>G18G41G90G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td>RT</td><td>G18G42G90G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td rowspan="2">YZ plane</td><td>LF</td><td>G19G41G90G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td>RT</td><td>G19G42G90G00(XxYyZz|XxZz)[AaBb]</td></tr>
<tr><td rowspan="2">APT</td><td colspan="3">point to point</td><td>RAPID;GOTO/(point|x,y,z)[,i,j,k]</td></tr>
<tr><td colspan="3">contour motion</td><td>GO/TO,line1,TO,pL1,(TO|ON|PAST), line2</td></tr>
<tr><td colspan="4">VAL II</td><td>MOVE (x,y,z,o,a,t|location)
MOVET (x,y,z,o,a,t|location,float)</td></tr>
<tr><td colspan="4">ASEA</td><td>ROBOT COORD
POS V=1% x,y,z,Q1,Q2,Q3,Q4</td></tr>
<tr><td colspan="4">UniSet</td><td>Goto,rapid[,-tool|0|tool].(location|
line1@line2@pl1| x[@y]@z[,@[@]]
[,offset|x@y@z])</td></tr>
</table>

(wrist coordinate)에 관한 방향을 정의하기 위해 quaternion Q1, Q2, Q3, Q4를 사용한다.

위에 논의한 Word Address, APT, VAL II, ASEA의 조인트 보간 운동을 위한 명령어의 syntax 분석으로 두 개의 UniSet 명령어 포맷이 정의되었다. UnaryModifiedCommand포맷인 Goto, rapid:()는 절대운동과 상대운동을 위해 각각 사용되어지며, BinaryModifiedCommand포맷인 Go, rapid, tool:()은 절대 운동 모드를 위해 사용된다. 한정어 tool은 APT언어에서의 한정어 {TO|ON|PAST}를 지원하기 위해 고려된 한정어이다.

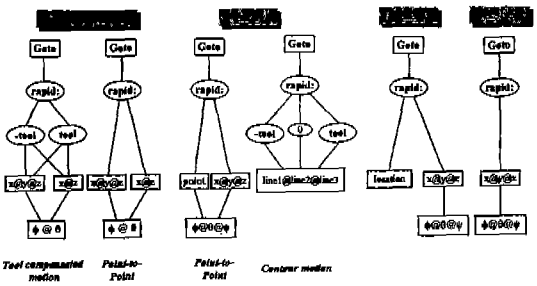

Fig. 3 UniSet Mapping to Target Languages for Joint Interpolation (Absolute Mode)

Table 3. Joint Interpolation: Relative Mode

Word Address	G91G00{X±Y±Z±}[AaBb]
APT	RAPID:GODLTA/x,y,z[,i,j,k]
VAL II	APPRO (location|(x,y,z,o,a,t),offset) DEPART offset
ASEA	ROBOT COORD POS V=[% x,y,z,Q1,Q2,Q3,Q4 OFFSET x,y,z MM
UniSet	Go,rapid:{location|offset}[x[@y]@z[, @ [@]]],[,z@y@z]

표 2와 3에서 UniSet 명령어의 변수들을 위해 서술적인 이름들(예: line1, line2 등)이 사용되고 있는데, 일단 목표 기계가 지정되면 객체 지향언어의 다중성의 특징을 이용하여 변수들의 이름을 해당 기계언어의 포맷에 맞게 변환되어 진다. UniSet의 조인트 보간 운동의 절대 모드에 대한 자동 언어들과의 명령어, 한정어, 변수들의 순서와 일대일 대응을 나타내는 도표가 그림 3에 나타나 있다.

UniSet의 모든 변수들은 ASEA 로봇의 변수들을 제외한 자동 언어들의 변수에 바로 일대일 대응된다. ASEA 로봇이 엔드 에펙터의 자세를 나타내는데 quaterion을 사용하는 반면 UniSet은 로봇좌표계를 참조하는 오일러각(φ, θ, δ)을 사용한다. 따라서 오일러각(φ, θ, δ)으로 부터 quaterion으로의 변환이 요구된다.

3.2.2 선형 보간 운동 명령어

선형 보간 운동은 모든 축 드라이버들이 정확한 경로를 만들어내기 위해 조정 하면서 로봇의 엔드 에펙터를 또는 NC 공작기계의 공구를 안내한다. 경로가 부분적으로 선형화되어 있거나 선의 좌표계가 명시되어 있다면 다른 특정 경로를 따른 운동도 가능하다.

표4와 5는 각각 절대모드와 상대모드의 선형 보간 운동에 대한 자동 언어들의 비교와 분석 결과의 UniSet명령어를 보여준다. Word Address는 코드 G01을 선형 보간 운동방식을 정의하기 위해 사용한다. 다른 모든 G 코드들은 선형 보간 운동이 적용되기 전에 기술되어야 하며, 선택적인 이송률 f가 명시되지 않는다면 이미 선언된 값들이 사용된다.

APT 언어는 명령어 RAPID:GOTO와 RAPIO:GODLTA가 조인트 보간 운동을 나타내는데 반해, RAPID가 없어질 때, 즉 명령어 GOTO와 GODLTA는 각각 절대 및 상대 운동 모드의 선형 보간 운동을 정의한다. 명령어 GOTO을 위한 위치와 자세를 위한 변수들은 조인트 보간 운동과 동일하다. APT의 조인트 보간 절대 모드의 명령어 GO가 가공 공정의 준비에 기술되었다면, 뒤따르는 명령어는 선형 보간 절대 운동 명령어 GOFWD, GOLFT, GORGT중의 하나이다.

Table 4. Linear Interpolation: Absolute Mode

Word Address	point to point			G90{G40|G01{X±Y±Z±|X±Z±}[AaBb][,F]}
	Tool compensated motion	XY plane	LF	G17G41G90G01{X±Y±Z±|X±Z±}[AaBb][,F]
			RT	G17G42G90G01{X±Y±Z±|X±Z±}[AaBb][,F]
		XZ plane	LF	G18G41G90G01{X±Y±Z±|X±Z±}[AaBb][,F]
			RT	G18G42G90G01{X±Y±Z±|X±Z±}[AaBb][,F]
		YZ plane	LF	G19G41G90G01{X±Y±Z±|X±Z±}[AaBb][,F]
			RT	G19G42G90G01{X±Y±Z±|X±Z±}[AaBb][,F]
APT	straight cut			GOTO/(point|x,y,z)[,i,j,k][,f]
	contour motion			(GOFWD,GOLFT,GORGT)/x1,(TO|ON|PAST),z[,f]
VAL II				MOVES {x,y,z,o,a,t|location} MOVEST {x,y,z,o,a,t|location,}[test]
ASEA				RECT COORD POS V=[% x,y,z,Q1,Q2,Q3,Q4
UniSet				Goto[,left|right][,- tool#|tool]:{location|x1@z[x[@y]@z[, ■ [@]]],[,offset|x@y@z#]

Table 5. Linear Interpolation: Relative Mode

Word Address	G91G01{X±Y±Z±|X±Z±}[AaBb][,f]
APT	GODLTA/(number|x,y,z)[AaBb][,f]
VAL II	MOVES HERE:{x,y,z,o,a,t|location} MOVEST HERE:{x,y,z,o,a,t|location,}[test]
ASEA	RECT COORD POS V=[% x,y,z,Q1,Q2,Q3,Q4 OFFSET x,y,z MM
UniSet	Go:{location|x[@y]@z[, @ [@]]},[,offset|x@y@z#]

선형 보간 운동을 위한 VAL II 명령어는 절대 모드를 위해서는 명령어 MVES와 MOVEST, 상대모드를 위해서는 명령어 MOVES HERE와 MOVEST HERE가 있다. 조인트 보간 운동의 경우와 유사하게, 각 모드의 두 번째 포맷이 운동 중인 엔드 에펙터의 상태에 영향을 주는 서식 float를 사용하여 열림 정도를 정의한다.

ASEA 로봇은 선형 보간과 조인트 보간 운동을 위해 같은 명령어를 사용한다. 앞에서 언급한 것처럼, 명령어 ROBOT COORD는 모드를 조인트 보간 운동에 맞추고, 명령어 RECT COORD는 모드를 선형 보간 운동에 맞춘다.

UniSet의 선형 보간 운동을 위한 명령어는 Goto(절대 모드)와 GO(상대 모드)인데, 운동의 방향을 결정하기 위해 한정어 left와 right가 사용된다. 명령어를 위한 변수들은 이송율 f가 첨가되는 것을 제외하고 조인트 보간 운동을 위한 변수들과 일치한다

4. UniSet을 이용한 제조셀 프로그램

셀 프로그램은 로봇 프로그램, NC공작기계의 파트 프로그램, 순차적인 작업 수행을 정의하는 제어 프로그램으로 구성되어 있다. 유연제조셀 프로그램의 일반적인 구조는 적절한 센서 신호들에 의해 동기화되는 해당 기계들의 수행을 위한 부분적인 프로그램들의 집합이다. 이러한 부분 프로그램들의 일부는 동시에 수행될 수도 있다.

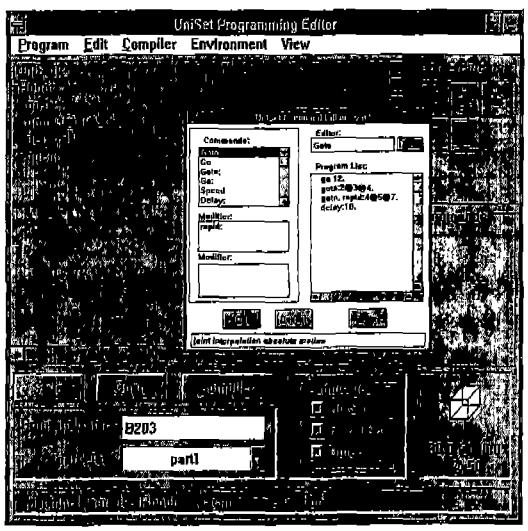

Fig. 4. User Interface for the Cell Programming Editor

UniSet 접근법에서, 셀 프로그램은 동기화된 신호들에 의해 상호 작용을 하는 프로그램 부분들을 시리얼 형태로 개발할 수 있다. 이러한 시리얼 코드 개발의 접근법은 매우 자연적이고 직관적이며, 유연제조셀 프로그램을 매우 간단히 만든다. 그림 4는 셀 프로그래밍 모듈의 사용자 그래픽 인터페이스를 나타낸다. 단일 언어의 개념의 도입으로 셀 프로그래밍 작성에 있어서 각각의 기계들의 언어 모두를 사용하는 대신 하나의 언어에만 집중 함으로서 시간적 절약뿐만 아니라, 프로그래머의 수를 줄일 수 있는 이점을 안겨준다. 또한 셀 프로그램 개발은 단일 언어 혹은 해당 기계의 고유 언어들을 이용하여 개발하는 혼합 언어(mixed languages)방식을 택하였다. 이러한 접근법은 셀 구성 기계고유의 특징들이나 특정 언어에서의 셀 프로그래머의 경험을 최대한 활용 할 수 있다.

5. UniSet코드의 해당기계 코드로의 번역

유연제조셀 내에서 주어진 작업을 수행하기 위해 필요한 셀 프로그램이, UniSet으로 개발되었다면, 해당기계에서 각각의 작업을 실행하기 전에 기계들이 실행할 수 있는 언어의 명령어 형태로 변환되어야 한다. 번역 과정에서는 명령어들과 변수(argument)들은 따로 고려되어졌다. 인공지능 언어인 prolog를 이용하여 UniSet과 해당 기계 언어간의 명령어들을 번역하기 위해 사용된다. 일대일 대응을 나타내는 서식들이 prolog에서 predicates의 변수들로 표현되는데, 그림 5는 predicate에 의한 번역 룰들이 보여진다.

그림을 참조하면, 형식이 UnaryCommand와 Command인 명령어들의 번역은 UniSet과 기계 고유의 명령어들 사이에서 상대적으로 단순한 일대일 대응이 존재한다. 예를 들면, 로봇이 gripper를 닫는 것을 서술하는 UniSet 명령어 close는 VAL II 명령어 Close로 번역된다. 이에 반해, 한정어를 가지는 ModifiedUnaryCommand, UnaryModifiedCommand, BinaryModifiedCommand 형식의 명령어들은 한정어의 문맥에 종속되어 번역된다. 예를들면, UniSet 명령어 coolant의 번역은 한정어 on과 off에 종속하고, on에 대응하여 Word Address 명령어로는 M08, off 한정사에 대응하여 명령어 M09라는 결과를 이끌어낸다.

객체지향 언어인 Smalltalk의 syntax 사용 때문에, UniSet으로 표현된 변수들의 형식은 기계고유 언어의 형식들과는 다르다. 따라서, 변수를 해당기계의 형식으로

다시 만드는 것은 번역 모듈에 의해 수행되어진다. 변수들의 변환은 명령어의 번역과 달리 번역 메카니즘에에 의해 수행되기 보다는 각각 UniSet 명령어의 객체에 의해 수행되도록 한다. 이 방법은 명령어과 관련된 각 변수들이 UniSet 명령어 객체에 포함된 지식을 바탕으로 자동적으로 번역하도록 한다. VAL II, APT 그리고 Word Adress의 경우, 변수들의 변환은 상대적으로 간단하다. 그러나, 앞에서 언급한 것처럼 UniSet은 자세를 서술하기 위해 로봇 좌표계를 참조한 오일러 각도를 사용하는 반면에, ASEA 로봇은 엔드 에펙터에 부착된 손목 좌표계에 따라 자세를 정의하기 위해 quaternion을 사용한다. 결과적으로 오일러 각도로부터 quaternion으로의 변환 과정은 UniSet 명령어 객체들의 상호관계를 근거로한 Smalltalk에 정의된 객체에 의해 수행된다.

6. UniSet접근법의 실용성 검사

여기서는 UniSet으로 부터 자동으로 생성된 해당기계 프로그램의 실용성을 검증하기 위해 해당기계언어를 이용하여 개발된 프로그램과 비교하였다. 예로서 도입된 작업은 로봇, NC기계, 컨베어로 구성된 유연제조셀 내에서 로봇을 이용하여 부품을 컨베어에서 집어서 NC기계에 투입하는 "pick/load" 작업이 고려되었고, 해당기계 자동언어로는 VAL II가 사용되었다. 그림 6에서 주어진 제조셀 환경에서 "pick/load" 작업의 수행을 위해 UniSet으로 개발된 셀 프로그램을 예시하였는데, 일부 수치적인 정보가 셀 프로그램 단계에서는 숨어있다. 시스템에 포함된 번역 모듈은 로봇 좌표계에 대한 로봇의 엔드 에펙터의 위치들을 생성해낸다.

```
!ExpertUniTranslator methods !

trans('goto','rapid','MOVE','val2','UnaryModifiedCommand').
trans('goto','rapid:','G00:G00','word','UnaryModifiedCommand').
trans('goto:','POS','asea','Command').
trans('go ','G01:G01','word','Command').
trans('go:','GODLTA/','apt','Command').
trans('go:','MOVES HERE','val2','Command').
trans('go:','POS OFFSET','asea','Command').
trans('gocircle','G17G02','word', inaryModifiedCommand .
trans('spindle','ccw:','M04S','word','UnaryModifiedCommand').
trans('spindle','ccw:','SPINDL/,CCW','apt','UnaryModifiedCommand').
trans('coolant','off','COOLNT/OFF','apt','ModifiedUnaryCommand').
trans('coolant','off','M09','word','ModifiedUnaryCommand').
trans('coolant','on','M08','word','ModifiedUnaryCommand').
  ------
trans('close','M22','word','UnaryCommand').
trans('close','Close','val2','UnaryCommand').
trans('close','GRIPPER','asea','UnaryCommand').
trans('close:','close' 'val2','Command').
trans('tool:','G10H<num>Z<num>','word','Command').
```

Fig. 5. Prolog based Translation Rules

```
{UniSet}
Goto:pick_via      ;Move to the top of a part on a conveyor
Goto:pick          ;Move to a part
Close.             ;Grasp a part
Go:50.             ;Back off
Goto:conveyor      ;Return to the previous location

Goto:mill          ;move to a mill

Goto:load_via      ;Approach to the load position
Goto:load          ;Move to the load position
Open.              ;Release a part
Go:70.             ;Back off
Goto:mill          ;Return to the previous location
```

Fig. 6. UniSet Program for pick/load task

자동 생성된 코드와 VAL II를 사용하여 독립적으로 개발된 코드 사이의 비교는 Table 6에 나타나 있다. 로봇이 주어진 작업 수행을 위해 필요한 Mill, pick, load 등 중요한 위치들은 미리 정의되어야만 한다. 월드 모델링 기법을 이용하여 이러한 위치들을 정의할 수 있지만, 실용적인 측면에서는 이러한 변수들을 정의하는 편리한 방법은 로봇을 파워 리드(power lead)나 수동 리드(manual lead)를 사용하여 원하는 장소에 위치시키고 그 위치를 변수에 할당하고 메모리에 기록하는데, UniSet에서는 Position, VAL II에서는 HERE Mill 이라는 명령어가 로봇의 현위치를 변수명에 할당하기 위해 사용된다.

Table 6. Comparison generated codes and developed codes in VAL II

No	Generated Codes	Developed Codes	Comments
1	MOVES(150,140,120,30,40,30)	APPRO *pick*,30	Approach the part
2	MOVES(145,110,80,30,40,30)	MOVES *pick*	Move to the part
3	CLOSE	CLOSE	Grasp the part
4	DEPART 50	DEPART 50	Back off
5	MOVES(100,60,120,30,40,30)		Return to the previous location
6	MOVES(-120,150,150,0,90,10)	MOVES *Mill*	Move to the mill
7	MOVES(-120,200,110,0,90,10)	APPRO *load*,50	Approach to the load position
8	MOVES(-125,244,100,0,90,10)	MOVES *load*	Move to the load location
9	OPEN	OPEN	Release the part
10	DEPART 70	DEPART 70	Back off
11	MOVES(-120,160,150)	MOVES Mill	Return to the previous location

자동 생성된 코드에서 모든 목표 위치들은 제조셀 환경에 따라 계산된 수치적인 형식으로 표현된다. 첫 세개의 숫자들은 엔드 에펙터의 위치를 표시하고, 마지막 세 개 수는 자세(orientation)를 나타낸다. 이러한 특정 형식들은 번역모듈에 의해 UniSet 코드들로부터 번역되어진다.

Table에 의하면, 라인 번호 1과 7에서 VAL II를 이용한 프로그래머는 명령어 APPRO를 사용하는 반면, 번역모듈은 명령어 MOVES를 생성한다. 이것은 번역 모듈에

서 로봇이 NC기계에 접근하는데 객체 충돌회피 모듈에 의해 최적화된 경로를 정의하기 때문에 직선 움직임을 정의하는데 편리한 명령어 MOVES를 사용한다. 명령어 MOVES와 APPRO는 정확히 같은 로봇의 움직임을 일으킨다. 라인 번호 5에서 VAL II 프로그래머는 명령어를 사용하지 않는 반면에 번역 모듈은 명령어 MOVES를 생성시킨다. 여기서 다시 충돌 회피 모듈은 제조셀을 구성하는 기계들 사이를 로봇이 충돌 없이 항해하기 위해 중요한 위치들이 미리 정의되어 그러한 위치들을 꼭 지나도록 하였다.

7. 결 론

UniSet은 멀티-벤더 기계 통합과 운용에 관련된 문제들을 다루기 위해 유연제조셀을 위한 생산 지시 명령어 집합으로 개발되었다. UniSet은 제조 셀 및 구성 기계들이 다양한 부품을 제조하는데 요구되는 모든 동작들을 서술할 수 있도록 기본적인 명령어들을 포함하고 있다. 현재 존재하는 자동화 언어들의 비교 및 분석을 기반으로 하여 개발되었으며, 명령어들은 일관성 있는 준 영어 표현에 기반을 둔 6개의 포맷으로 분류되어진다. 이러한 UniSet을 이용하여 셀을 프로그램을 하면 해당기계 언어보다 단일한 명령어를 사용함으로써, 평범한 작업자가 셀을 프로그램 할 수 있고 시간을 절약할 수 있는 이점을 부여하고 있으며, UniSet 접근법의 다른 이점은 배제적인(exclusive) 접근법이 아니라는 점이다. 즉 셀 프로그래머는 UniSet으로 전체 셀을 프로그램할 수 있을 뿐만 아니라, 해당 기계 제어 언어들로 프로그램할 수 있어, 셀 프로그래머의 뛰어난 경험을 이용 할 수 있는 환경을 공급한다. 셀 프로그래밍을 쉽게 하기 위해 그래픽 사용자 인터페이스인 UniSet 에디터라는 프로그램 모듈이 문맥에 민감하도록 개발되었으며, 사용자는 해당기계들을 위한 명령어들의 리스트로부터 원하는 명령어들을 선택하여 셀 프로그래밍을 완성할 수 있다.

UniSet으로 개발된 셀 프로그램은 해당기계들에게 다운로드되어 실행되기 전에 각각 다른 고유기계 언어의 명령어들로 번역되어 진다. 이 번역 작업은 인공지능 언어 인prolog에 의한 번역 룰들을 사용하여 수행한다.

이 논문에서는 UniSet 번역 모듈은 오직 현존하는 네 개의 자동 생산언어(Word Address, APT, VAL II, ASEA)를 다루는 것에 한정되어 있다. 이 모듈은 다른 자동 언어들의 포함의 요구에 부응하기 위해 새로운 언어들을 추가하는 과정이 매우 간단하도록 고려되었다.

참 고 문 헌

1. Grossman, D.D., Short, M.M., "AML- much more than robot language", Technical paper presented at Robots 9 Conference, SME Technical paper No. MS85-612, June 2-6, 1985.
2. Grossman, D.D., "AML as a plant floor language", Robotics & Computer Integrated Manufacturing, Vol.2, No.3/4, pp.215-217, 1985.
3. Sharon, A., Carey, E., "CML+PC=CIM" Computers in Mechanical Engineering, Vol.4, pp. 30-34, July 1985.
4. Li,L., "Object-Oriented Modeling and Implementation of Control Software for a Robotic Flexible Manufacturing Cell", Robotics and Computer-Integrated Manufacturing, Vol.11, No.1, pp.1-12, March 1994.
5. Magy,E., Haidegger,G., "Object-Oriented Approach for analyzing, Designing and controlling Manufacturing cells" AUTOFACT'93, Chicago, USA, pp.MS244_1-10, November 8-11, 1993.