

◆특집◆ STEP을 기반으로 하는 CNC 기술

STEP-NC 구현을 위한 ISO 14649의 역할 및 발전 방향

한주영*, 장병열*, 조현보**, 서석환**

Roles and Visions of ISO 14649 for STEP-NC

Juyung Han*, Pyoung Yol Jang*, Hyunbo Cho**, Suk-Hwan Suh**

Key Words : ISO 14649, STEP-NC, CNC

1. 서론

NC 기술은 하드웨어적, 소프트웨어적으로 빠르게 발전하고 있다. 소프트웨어적인 측면에서 NC제어기는 다양한 적응제어, 지능제어를 지원하는 추세이며, NC 가공기의 경우 고속가공, 정밀가공의 기능적 측면에서 발전이 이루어지고 있다. 뿐만 아니라, 컴퓨터 기술의 발전에 힘입어 NC가공을 지원하기 위한 환경(CAD, CAM, CAE) 역시 빠르게 변화하고 있다. 그러나 CAM과 CNC 사이의 데이터 인터페이스는 여전히 ISO 6983("G-Code")이 사용되고 있다. ISO 6983은 축 동작 제어 중심의 표준으로서 제작사마다 자사 고유의 코드로 변형되어 사용되고 있다. 이는 직선과 원호 보간만을 지원하는 등의 문제점으로 인해 NC기술의 발전에 저해 요소가 되고 있다.

이러한 배경으로 1994년 유럽에서 시작된 CNC로의 데이터 인터페이스에 대한 새로운 대안을 찾는 연구로서 수행된 OPTIMAL 프로젝트는 제품데이터모델(STEP)을 기반으로 한 새로운 데이터 인터페이스의 가능성을 입증하였다. OPIMAL 프로젝트의 결과를 바탕으로 하여 1997년 CNC 데이터 인터페이스의 새로운 표준으로 ISO 14649 ("Data model for Computerized Numerical Control")가

제안되었고, 2000년 4월 현재 밀링 가공에 대한 공정 데이터 모델 표준은 DIS(Draft for International Standard)단계에 있다.¹⁾ 이 밖에 선반, 방전가공, 와이어 가공, RP(Rapid Prototyping) 등과 같은 가공에 대한 표준도 향후 제정될 예정이다.

ISO 14649는 제품데이터모델(STEP)을 기반으로 한, 미래 지향적 CNC 데이터 인터페이스로서, 기하 정보를 포함한 가공 피쳐 정보 중심의 인터페이스이다. 따라서 CNC는 고속 가공, 정밀가공의 기능을 충분히 활용할 수 있게 되며, 자동 공구 경로 생성, 이상 상황 대처와 같은 고도의 지능적 기능도 수행할 수 있게 된다. 또한 이미 표준으로 자리잡은 STEP을 그대로 수용하기 때문에 CAD, CAE 시스템과의 데이터 흐름이 보다 용이해 지는 특징을 가진다.

ISO 14649의 표준 제정 작업과 함께, 1997년 IMS 프로젝트로서 미국의 NIST가 중심이 된 "STEP-NC" 프로젝트가 시작되었다. STEP-NC는 ISO 14649를 표준 입력으로 하여 자율적, 지능적인 기능을 수행하는 CNC 환경을 말한다.

본 논문에서는 2장을 통해 최근의 CNC환경의 변화와 STEP-NC의 연구에 대해 살펴보고, STEP-NC의 데이터 인터페이스인 ISO 14649를 특징과 구조, 문법을 중심으로 3장에서 설명한다. 4장에서는 예제 파트를 통해 ISO 14649 형태의 NC 프로그램과 작업 시나리오를 정리하고, 5장을 통해 현 ISO 14649의 문제점을 파악하고 이에 대한 해결 방안을 제안한다.

* 포항공과대학교 기계산업공학부 대학원
** 포항공과대학교 기계산업공학부
Tel. 0562-279-8243, Fax. 0562-279-2870
Email hcho@postech.ac.kr

2. ISO 14649의 도입에 따른 변화

2.1 CNC 환경의 변화

현재 CNC 환경은 과거의 경직된 환경에서 벗어나 개방성, 상호 운용성, 확장성, 이식성, 자율성, 지능성 등을 요구하는 새로운 패러다임을 맞이하고 있다. 현재 CNC에는 이러한 환경 변화에 맞추어 기존의 한계를 뛰어넘는 다음과 같은 기술 변화에의 적응 필요성이 대두되고 있다.

첫째, 객체로써의 CNC이다. 과거에는 CNC 동작 기제는 가공의 주체로써 인식되었으나 현재는 MMS (Manufacturing Message Specification) 등의 부상으로 주체가 아닌 객체로 인식되고 있다. 아울러 CNC 동작 기제를 제품의 전체 Life Cycle(설계, 해석, 가공, 검사, 보수)에서의 하나의 부분으로써 인식하고 있다.

둘째, 파라메트릭 프로그래밍(Parametric programming)의 지원 기술이다. 파라메트릭 프로그래밍은 기존의 ISO 6983 프로그래밍은 형상과 작업이 비슷한 파트들의 가공에 적합하지 않아 그룹 테크놀로지(Group Technology) 등을 효율적으로 지원할 수 없다. 이를 극복하기 위하여 파라메트릭 프로그래밍에서는 C, Pascal 등에서 사용되는 변수, While 구문, If 문 등을 사용하여 기존의 ISO 6983 프로그래밍의 단점을 극복하고 있다.²⁾

셋째, CAD/CAE/CAPP/CAM/CAx 시스템들 사이의 데이터의 효과적인 교환을 위한 연구에 많은 노력이 기울여지고 있다.³⁾ 이를 위해서는 통합적 데이터 모델의 개발에 관한 연구가 있었으며, 인터넷과 STEP에 기반한 데이터의 교환 기술을 이용하여 가상기업에의 응용 분야에 대한 연구도 있었다.⁴⁾ 특히 STEP의 응용은 현재까진 초보적인 단계이지만 앞으로 STEP을 이용한 데이터의 교환 기술의 중요성은 더 커질 것이다.

2.2 STEP-NC

원래 STEP 정보는 모든 시스템에서 사용될 수 있도록 고안되었으나, 실제로 STEP 정보는 CAD 시스템과 CAM 시스템과의 상호 교환은 가능하나, CAM 시스템과 CNC 시스템과의 STEP 정보의 교환은 이루어지지 않아 전체 시스템의 성능 향상에 저해 요소가 되었다. 대신에 기계의 축 동작 및 기계 조작에만 국한된 ISO 6983에 기반한 매우 경직된 데이터가 CAM 시스템에서 CNC 시스템으로

넘어가게 되어 새로운 CNC 환경에의 적응에 많은 제약을 가하게 되었다.

이러한 제약을 극복하기 위하여, STEP-NC에 관한 연구가 진행중이다. STEP-NC는 STEP에 기반한 표준 제품 모델 및 가공정보를 입력 받아, ISO 6983의 문제점을 해결하고 다양한 지적 기능을 수행할 수 있는 차세대 CNC이다. STEP-NC에 필요한 데이터 모델은 ISO TC184/SC1에서 ISO 14649의 문헌으로 국제 표준화가 이루어 지고 있다. STEP-NC는 현재 제정 중인 ISO 14649를 데이터 인터페이스로 하며, 자동 공구 경로 생성, 자동 공구 선정, 자동 절삭조건 제공, 이상상태감지 및 복구, 응급상황에 대한 자율적 대처, 가공상태 및 결과 피드백 등을 구현하는 CNC 시스템으로서 변화하는 CNC 환경에 효율적으로 적용할 수 있을 것이다.

3. ISO 14649의 개요

3.1 특징

현재까지 CNC데이터 인터페이스의 표준으로 사용되고 있는 ISO 6983("G-Code")는 축 동작 및 기타 NC 가공기의 기능을 제어하기 위한 것을 목적으로 하는 언어이다. OPTIMAL 프로젝트의 최종 리포트에서는 ISO 6983의 단점들을 다음과 같이 정리하였다.⁵⁾

- 직선(G01), 원(G02, G03) 운동만 지원하여 고품위 가공이 어렵다.
- 제품에 대한 기하학적 정보를 가지지 못하여 5축 가공이나 고속가공에 부적합하다.
- 공정계획 정보를 가지고 있지 않다.
- 가공정보를 가지고 있지 못하여 최적화된 가공조건이 CAD/CAM으로 피드백되지 못한다.
- 자사 고유 코드를 가지는 경향으로 인해 데이터 교환이 어렵다.

이러한 단점들을 극복하기 위해 제안된 ISO 14649는 다음과 같은 특징을 가지는 것을 목적으로 제정 중이다.

- 피쳐 기반의 NC 프로그램을 통해 수정이 용이하다
- 제품에 대한 기하학적 정보(STEP)를 별도의

변환 가정 없이 사용한다.

- 스플라인 데이터를 통한 가공으로 고품위 가공을 가능하게 한다.
- 가공정보, 공정계획 정보를 가지고 있어 CNC로 하여금 자동 공구경로 생성 및 수정, 자동 공구 선정, 자동 절삭조건 제공 등의 지능적 제어를 가능하게 한다.

피쳐 기반의 특징과 함께 ISO 14649에서는 “워킹스텝(Workingstep)”이란 새로운 개념을 도입하였다. 워킹스텝은 공구의 이송과 가공에 대한 기본 단위이며, NC 프로그램에서 수행 가능한 유일한 데이터이다. 각 워킹스텝은 가공을 위한 파라미터들과 가공대상이 되는 피쳐, 가공에 사용할 공구 등을 지정하게 되어 있다.

3.2 구성

ISO 14649는 여러 개의 Part로 구성된다. Part 1에서는 ISO 14649의 기본적인 개념을 소개하고, Part 2에서는 NC 프로그램을 작성하는 문법에 대한 EBNF(Extended Backus Normal Form)을 정의한다. Part 3에서는 NC 프로그램을 작성할 때 사용할 수 있는 선정의 함수(Prefined functions)를 설명하고 있으며, Part 9는 ISO 14649표준과 관련된 용어들을 정리하였다. 실제 공정 데이터 모델은 Part 11부터 Part 19까지 정의된다. 공정 데이터 모델은 STEP의 응용 프로토콜(Application Protocol)을 정의하는 표준 언어인 STEP Part 11의 EXPRESS로 정의되며, 공정 기술을 표현하기 위한 클래스 계층 구조라고 할 수 있다. 포켓(Open_pocket, Closed_pocket), 슬롯(Slot) 등과 같은 피쳐 클래스들과 포켓 황삭(Pocket_rough_manuf_data), 포켓 정삭(Pocket_finishing_manuf_data) 등의 워킹스텝 클래스들이 정의되어 있다. 현재는 Part 11과 Part 11/1이 정의되어 있으며, 이는 밀링에 대한 공정 데이터와 이에 대한 공구정보를 정의한다. 그리고 Part 12는 터닝 공정에 해당하며, 앞으로 그 밖의 방전가공, RP, Grinding과 같은 공정 데이터 모델에 대한 표준이 제정될 것이다.

각 공정 데이터 모델 표준은 STEP의 통합 응용 자원(Integrated-Application Resources)인 Part 42과 Part 43을 참조하며, AP 224(“Mechanical product definition for process planning using machining features”)에 정의된 가공 피쳐를 차용한다.

3.3 문법 구조

ISO 14649는 NC 프로그램의 작성에 있어 주 프로그램, 기술 정보, 기하 정보의 세 부분으로 나누어 기술하도록 하고 있다. 이렇게 함으로써 각 부분은 상호 참조의 형태를 가지게 된다. 프로그램의 구체적인 문법은 ISO 14649 Part 2에 정의되어 있으며, 물리적 저장 형태는 아스키 코드로 이루어진 텍스트 파일이다.

3.3.1 주 프로그램(Main Program)

주 프로그램은 기술 정보를 기술하는 부분에서 정의된 워킹스텝들과 기하 정보를 기술하는 부분의 기하 정보를 참조하여 CNC가 직접 수행할 수 있는 실행 구문들의 순서를 결정하는 부분이다. 주 프로그램의 문법은 프로그램 작성 언어인 C, C++와 흡사하여, 정수형, 실수형 등의 데이터 형을 지원하며, For문, If문과 같은 제어 및 반복 구문을 지원한다. 또한 ISO 14649 Part 3에 정의된 선정의 함수를 이용할 수 있으며, 임의로 함수를 작성하여 사용할 수도 있다. 주로 워킹스텝 단위로 NC프로그램이 작성되기 때문에 피쳐 단위에서의 프로그램 수정을 용이하게 한다. 가공을 실행하는 것은 “EXEC” 키워드를 이용하여 해당 워킹스텝을 지정하는 것으로 이루어진다.

3.3.2 기술 정보(Technology Description)

기술 정보 부분에서는 밀링, 터닝 등과 같은 공정과 관련된 세부 공정 단위들을 기술한다. 기술 정보는 가공 피쳐 정보, 단위 가공 정보, 가공시의 공작 기계 설정(절삭유 정보, 클램핑 축 설정 등)으로 크게 나뉜다. 이 때 각 가공 피쳐 정보는 기하 정보 부분에서 해당 기하 정보를 참조하여 기술된다. 단위 가공 정보는 각 가공 피쳐에 하나 혹은 여러 개의 워킹스텝을 할당하여 사용할 수 있다. 가공 피쳐와 워킹스텝을 선언하는 것은, 프로그래밍 언어에서 변수를 선언하는 것과 비슷하게 ISO 14649 Part 11~19에서 정의된 각 클래스(엔티티)에 필요한 속성값을 주는 식으로 이루어진다. Fig. 1은 ISO 14649 Part 11에 정의된 밀링 가공 피쳐인 포켓의 정의를 EXPRESS-G로서 보여주고 있다.

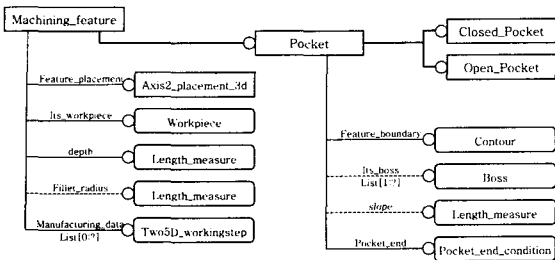


Fig. 1 EXPRESS-G schema of ISO 14649 Part 11

3.3.3 기하 정보(Geometry Description)

기하 정보는 STEP을 지원하는 Pro-Engineer™, SolidWorks™와 같은 CAD시스템에서 생성된 STEP 파일이다. STEP AP203과 AP224의 형식으로 사용이 가능하다. CAD시스템과의 데이터 호환성을 위해 기하 정보는 수정하지 않는 것을 원칙으로 하고 있다. 그러나 이러한 원칙을 따를 경우 유연한 공정계획 표현에 문제가 있어 수정할 필요가 있을 수 있다. 이 문제에 대해서는 5장에서 다루도록 한다.

4. ISO 14649 의 예제

4.1 예제 및 작업 시나리오

Fig. 2에서는 포켓(Pocket)과 홀(Hole) 등 두 개의 가공 피처를 갖는 파트에 대해서 ISO 14649를 따르는 NC 프로그램의 일부를 보여주고 있다. 주

프로그램에서는 기술 정보 부분에서 선언한 Pocket1_plunge, Pocket1_rough, Pocket1_sides_finish 등의 워킹스텝을 순차적으로 실행하도록 되어 있으며, 기술 정보 부분에서는 워킹스텝과 함께 가공 파트의 피처로서 Pocket1과 Hole1등을 선언한다. 피처를 선언할 때에는 STEP파일에서 해당 기하 정보를 참조한다. 이와 같이 ISO 14649는 피처를 중심으로, 각 피처에 적용하는 워킹스텝의 순서로서 NC 프로그램을 작성하도록 하여, 가공 시에 작업자는 주 프로그램의 판독이 쉬워지고, 가공의 순서를 쉽게 바꿀 수 있게 된다. 그리고 GUI를 지원하는 CNC 같은 경우, 특정 워킹스텝에 해당하는 피처를 그래픽으로 보여줄 수 있어 작업자는 NC 프로그램의 오류를 쉽게 찾을 수 있고, 이에 대한 피드백이 용이한 장점이 있다.

4.2 ISO 14649 형식의 NC 프로그램 예제

Fig. 2에 사용된 NC 프로그램의 전체 코드를 Fig. 3 에 나타내었다. 여기서 프로젝트 파일은 각 부분이 서로 다른 파일로 구성되었을 경우에 각 파일들의 이름을 명시하는 역할을 한다. Fig. 3 에 사용된 STEP 파일은 AP203 의 스키마를 따르며, 기술 정보 부분에서 참조 되는 부분을 제외한 내용을 생략하였다. 기술 정보 부분에서도 공구에 대해 선언한 내용도 일부 생략하였다.

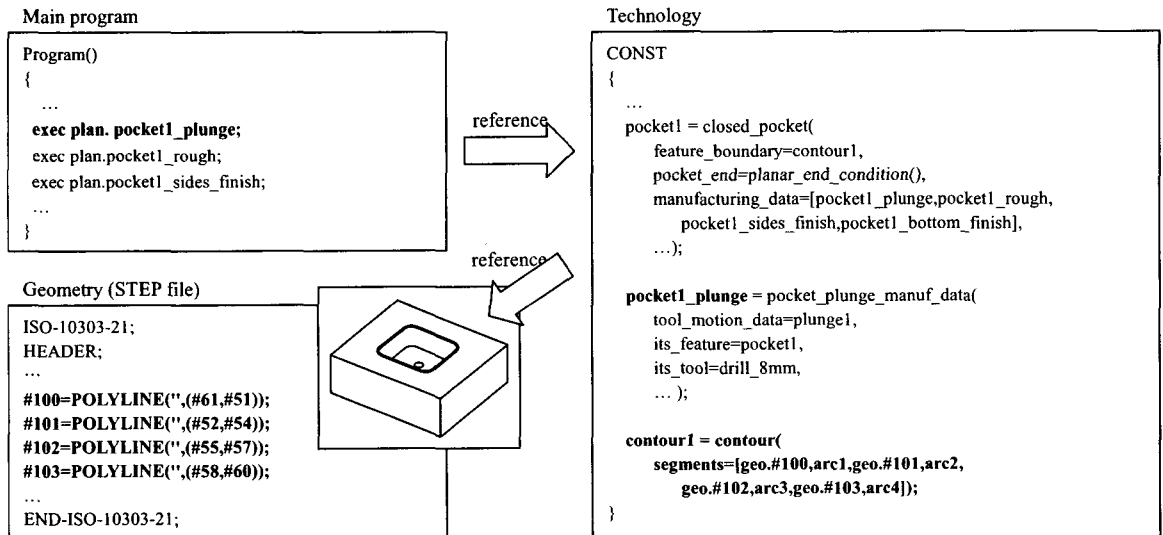


Fig. 2 NC program for the example part

A. Project File - "examp1.prj"

```
NcData_Ref = "c:\demo\NcFile.prg"
Plan = "c:\demo\Examp1.dat"
Geo = "c:\demo\Examp1.stp"
```

B. Main Program - "NcFile.prg"

DESCRIPTION

```
{
  filename = "c:\demo\NcFile.prg";
  progame = "EXAMP1";
  date = "04-10-2000";
  author = "POSTECH";
  comment = "Example of the ISO 14649 program";
}
```

CONST

```
{
  setup=axis2_placement_3d(location=cpt);
  cpt=cartesian_point(coordinates=[200.00,0.00,50.0]);
  withdrawalplane=elementary_surface(
    position=placement1);
  placement1=axis2_placement_3d(
    location=cartesian_point(
      coordinates=[0.00,0.00,70.00]));
}
```

PROGRAM() // MAIN PROGRAM

```
{
  set_placement(setup);
  set_secplane(withdrawalplane);

  exec plan.pocket1_plunge;
  exec plan.pocket1_rough;
  exec plan.pocket1_sides_finish;
  exec plan.pocket1_bottom_finish;
  exec plan.hole1_drill;
}
```

C. Technology Description - "Examp1.dat"

DESCRIPTION

```
{
  filename = "c:\demo\examp1.dat";
  progame = EXAMP1;
  date = "04-10-2000";
  author = "POSTECH";
  comment = "Example of ISO 14649 program";
}

CONST
{
  //*****workpiece*****
  examp1_workpiece = workpiece(
    manufacturing_features = [pocket1, hole1],
    its_material = material( id = 'St-50'));

  //*****features*****
  pocket1 = closed_pocket(
    feature_boundary=contour1,
    pocket_end=planar_end_condition(),
    feature_placement=placement2,
    depth=length_30,

    manufacturing_data=[pocket1_plunge,pocket1_rough,
      pocket1_sides_finish,pocket1_bottom_finish],
    its_workpiece=examp1_workpiece,
    id='CLOSED_POCKET');

  hole1 = round_hole(
    diameter=length_10,
    bottom_condition=through_bottom_condition,
    feature_placement=placement3,
    depth=length_20,
    manufacturing_data=[hole1_drill],
    its_workpiece=examp1_workpiece,
    id='HOLE1');

  //*****workingsteps*****
  pocket1_plunge = pocket_plunge_manuf_data(
    tool_motion_data=plunge1,
    its_feature=pocket1,
    its_tool=drill_8mm,
    its_technology=techno_drill1,
    its_machine_functions=machine_fn1,
```

Fig. 3 NC Program of ISO 14649

```

its_id='PLUNGE' );

plunge1 = spot_drilling_manuf_data(
  retract_plane=plane1,
  its_feature=pocket1,
  its_tool=drill_8mm,
  its_technology=techno_drill1,
  its_machine_functions=machine_fn1,
  its_id='PLUNGE_' );

pocket1_rough = pocket_roughing_manuf_data(
  tool_motion_data=rough1,
  its_feature=pocket1,
  its_tool=mill_8mm,
  its_technology=techno_mill1,
  its_machine_functions=machine_fn1,
  its_id='ROUGH' );

rough1 = bottom_and_side_rough_milling(
  finishing_allowance_side=len_allowance,
  finishing_allowance_bottom=len_allowance,
  its_feature=pocket1,
  its_tool=mill_8mm,
  its_technology=techno_mill1,
  its_machine_functions=machine_fn1,
  its_id='ROUGH_' );

pocket1_sides_finish
= pocket_sides_finishing_manuf_data(
  tool_motion_data=side_finish,
  its_feature=pocket1,
  its_tool=mill_8mm,
  its_technology=techno_mill2,
  its_machine_functions=machine_fn1,
  its_id='FINISH1' );

side_finish = side_finish_milling(
  its_feature=pocket1,
  its_tool=mill_8mm,
  its_technology=techno_mill2,
  its_machine_functions=machine_fn1,
  its_id='FINISH1_' );

```

```

pocket1_bottom_finish
= pocket_bottom_finishing_manuf_data(
  tool_motion_data=bottom_finish,
  its_feature=pocket1,
  its_tool=mill_8mm,
  its_technology=techno_mill2,
  its_machine_functions=machine_fn1,
  its_id='FINISH2' );

bottom_finish = plane_finish_milling(
  tool_motion_data=bottom_finish,
  strategy=contour_parallel(),
  its_feature=pocket1,
  its_tool=mill_8mm,
  its_technology=techno_mill2,
  its_machine_functions=machine_fn1,
  its_id='FINISH2_' );

hole1_drill = spot_drilling_manuf_data(
  retract_plane=plane2,
  its_feature=hole1,
  its_tool=drill_10mm,
  its_technology=techno_drill1,
  its_machine_functions=machine_fn1,
  its_id='DRILL1' );

//*****planes and placements*****
length_30 = length_measure(
  theoretical_size= 30.000 );
length_10 = length_measure(
  theoretical_size=10.000,
  upper_bound=10.01 );
len_allowance = length_measure(
  theoretical_size=0.50 );

contour1 = contour(
  segments=[geo.#100,arc1,geo.#101,arc2,
            geo.#102,arc3,geo.#103,arc4]);
arc1 = circular_arc(
  center_point=geo.#50,
  starting_point=geo.#51,
  end_point=geo.#52);

```

Fig. 3 NC Program of ISO 14649 (continued)

```

arc2 = circular_arc(
    center_point=geo.#53,
    starting_point=geo.#54,
    end_point=geo.#55);
arc3 = circular_arc(
    center_point=geo.#56,
    starting_point=geo.#57,
    end_point=geo.#58)
arc4 = circular_arc(
    center_point=geo.#59,
    starting_point=geo.#60,
    end_point=geo.#61);
placement2 = axis2_placement_3d(
    location=geo.#70);
placement3 = axis2_placement_3d(
    location=geo.#71);

plane1 = elementary_surface(
    position=placement4 );
placement4 = axis2_placement_3d(
    location=point1 );
point1 = cartesian_point(
    coordinates=[0.00, 0.00, 55.00]);

plane2 = elementary_surface(
    position=placement5 );
placement5 = axis2_placement_3d(
    location=point2 );
point2 = cartesian_point(
    coordinates=[0.00, 0.00, 25.00]);

*****technology*****
techno_drill1 = technology(
    feedrate = 0.10,
    spindle = 800.0);
techno_mill1 = technology(
    feedrate = 0.40,
    spindle = 4000.0);
techno_mill2 = technology(
    feedrate = 0.40,
    spindle = 2000.0);

*****machine_functions*****

machine_fn1 = machine_functions(
    coolant = true,
    chip_removal = true);

*****tools*****
drill_8mm = tool( ... );
drill_10mm = tool( ... );
mill_8mm = tool( ... );
}

D. Geometry Description - "Examp1.stp"

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('', '2;1');
FILE_NAME('examp1.stp','2000-04-07',(''),('',' ',' ');
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;

DATA;
...
#50=CARTESIAN_POINT(",(50.,40.,50.));
#51=CARTESIAN_POINT(",(40.,40.,50.));
#52=CARTESIAN_POINT(",(50.,30.,50.));
#53=CARTESIAN_POINT(",(100.,40.,50.));
#54=CARTESIAN_POINT(",(100.,30.,50.));
#55=CARTESIAN_POINT(",(110.,40.,50.));
#56=CARTESIAN_POINT(",(100.,60.,50.));
#57=CARTESIAN_POINT(",(110.,60.,50.));
#58=CARTESIAN_POINT(",(100.,70.,50.));
#59=CARTESIAN_POINT(",(50.,60.,50.));
#60=CARTESIAN_POINT(",(50.,70.,50.));
#61=CARTESIAN_POINT(",(40.,60.,50.));
#70=CARTESIAN_POINT(",(75.,50.,50.));
#71=CARTESIAN_POINT(",(75.,50.,20.));
...
#100=POLYLINE(",(#61,#51));
#101=POLYLINE(",(#52,#54));
#102=POLYLINE(",(#55,#57));
#103=POLYLINE(",(#58,#60));
...
ENDSEC;
END-ISO-10303-21;

```

Fig. 3 NC Program of ISO 14649 (continued)

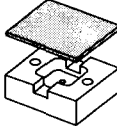
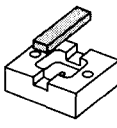
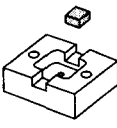
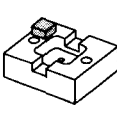
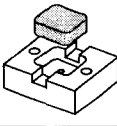
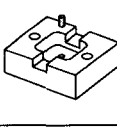
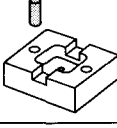
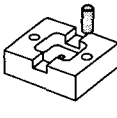
5. 문제점 및 해결방안

ISO 14649는 CNC 및 공작 기계의 향상된 기능에 초점이 맞추어져 있다. 기하 정보뿐만 아니라 공정계획정보를 추가적으로 포함하여, CNC가 자체적으로 공구경로를 생성하거나 기하 정보를 참고하여 기존의 직선 및 원호에 의존한 보간이 아닌 스플라인 보간과 같은 기능으로 고품위 가공을 가능하게 하는 등 향상된 기능을 지원하는 것을 목적으로 하고 있다. 그러나 현재까지는 표준 제정 단계에 있기 때문에 공정 계획 표현상의 문제, 구조적인 문제 등 몇 가지 문제점을 가지고 있다. 이러한 문제점들을 살펴보고 이에 대한 해결방안을 제안한다.

5.1 공정 계획 표현 상의 문제점

ISO 14649는 주 프로그램에서 각 워킹스텝의 순서가 결정된다. 워킹스텝의 순서는 공정 계획이라 할 수 있다. 그러나 워킹스텝의 순서는 순차적 선형 관계로만 표현이 가능하기 때문에 피쳐 그래프와 같은 비선형 공정계획(Non-linear Process Plans:NLPP)의 표현이 불가능하다. Fig. 4의 파트는 4장에서 사용된 예제에 몇가지 Table 1과 같은 가공 피쳐들을 추가한 것으로 Fig. 5에 나타난 피쳐 그래프를 따라 가공할 수 있다. 그러나 ISO 14649에서 표현 가능한 공정계획은 유연성이 부족하여 피쳐 그래프와 같은 대안 공정계획을 표현할 수 없다. 또한 Table 1의 F2(슬롯)은 F3, F4(포켓)을 대체 할 수 있는 가공 피쳐이다. 이러한 경우 슬롯을 표현하기 위해 필요한 기하 정보로 ISO 14649가 정의하고 있는 슬롯의 윤곽에 대한 정보가 기하 정보 부분에 나타나지 않을 수 있다. 따라서 기하 정보 부분의 STEP파일을 CAD시스템의 출력

Table 1 Machining features for the example part shown in Fig. 4 with workingsteps associated to them

No.	Features	Manufacturing Features Associative Workingsteps
F1		planar_face planar_face_roughing_manuf_data planar_face_finishing_manuf_data
F2		Slot slot_roughing_manuf_data slot_sides_finishing_manuf_data slot_bottom_finishing_manuf_data
F3		open_pocket pocket_roughing_manuf_data pocket_sides_finishing_manuf_data pocket_bottom_finishing_manuf_data
F4		open_pocket pocket_roughing_manuf_data pocket_sides_finishing_manuf_data pocket_bottom_finishing_manuf_data
F5		closed_pocket pocket_plunge_manuf_data pocket_roughing_manuf_data pocket_sides_finishing_manuf_data pocket_bottom_finishing_manuf_data
F6		round_hole spot_drilling_manuf_data
F7		round_hole multistep_drilling_manuf_data boring_manuf_data
F8		round_hole multistep_drilling_manuf_data boring_manuf_data

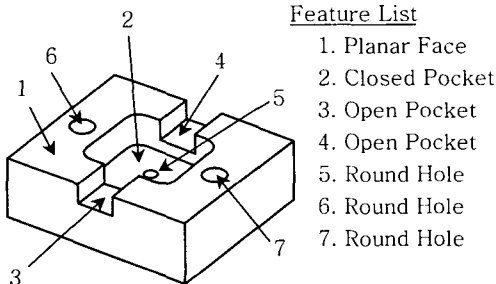


Fig. 4 Example part and its machining features

Feature List

1. Planar Face
2. Closed Pocket
3. Open Pocket
4. Open Pocket
5. Round Hole
6. Round Hole
7. Round Hole

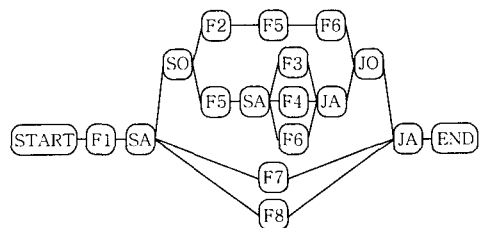


Fig. 5 Possible feature graph representing alternative process plans for the example part shown in Fig. 4

그대로 사용하도록 하는 현재의 원칙은 수정될 필요가 있다.

비선형 공정계획과 ISO 14649 를 접목하기 위한 대안으로는 ISO 14649 의 내부적 수정과 외부적 보완의 두 가지를 생각할 수 있다. 내부적 수정안으로는 주 프로그램 문법을 보완하여 지금의 순차적인 워킹스텝 표현과 함께 피쳐 그래프와 같은 공정 계획을 표현할 수 있도록 하는 것이다. 이러한 방법을 채용할 경우 CNC 는 비선형 공정 계획을 인식하고 직접 스케줄링할 수 있는 능력을 갖추어야 할 것이다. STEP-NC 는 이러한 능력을 갖는 CNC 에 대한 연구를 포함하고 있다. 외부적 보완책은 ISO 14649 의 문법은 그대로 두고 ISO 14649 를 지원하는 생산시스템의 구성을 Fig. 6 과 같이 하여 비선형 공정계획으로부터 온라인 상태에서 스케줄링하여 결정된 순차적 공정계획을 ISO 14649 의 문법에 맞도록 생성하는 방법이다.

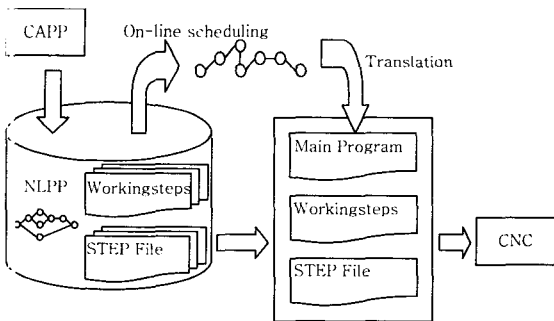


Fig. 6 On-line scheduling and generating ISO 14649

5.2 스키마 구조의 문제점

밀링 스키마(ISO 14649 Part 11)를 분석한 결과, 정보가 중복되어 사용되고 있음을 발견하였다. 문제가 되는 클래스는 Feature_manuf_data의 하위 클래스들인 아래의 클래스들이다.

- Pocket_plunge_manuf_data
- Pocket_roughing_manuf_data
- Pocket_sides_finishing_manuf_data
- Pocket_bottom_finishing_manuf_data
- Planar_roughing_manuf_data
- Planar_finishing_manuf_data
- Profile_roughing_manuf_data
- Profile_finishing_manuf_data
- Step_roughing_manuf_data
- Step_side_finishing_manuf_data
- Step_bottom_finishing_manuf_data
- Slot_roughing_manuf_data

- Slot_sides_manuf_data
- Slot_bottom_manuf_data
- Transition_manuf_data

각 클래스는 해당 피쳐와 연관되어 가공 조건들에 대한 정보로서 선택적인 속성들을 제외하면 Table 2와 같은 속성들을 가진다.

Table 2 Attributes of Pocket_roughing_manuf_data

Attributes	Type
Its_id	Label From 'Workingstep'
Its_feature	Manufacturing_feature
Its_tool	Tool
Its_technology	Technology
Its_machine_functions	Machine_functions From 'Milling_workingstep'
Tool_motion_data	Bottom and side rough milling

여기서 Tool_motion_data는 Generic_manuf_data의 하위 클래스 중의 하나와 연관된다. 그러나 Fig. 7에서 보듯이 Generic_manuf_data도 Milling_workingstep의 하위 클래스이기 때문에 Table 2에서의 Its_id, Its_feature, Its_tool과 같은 속성을 모두 가지고 있으며, 이들 속성들은 Feature_manuf_data의 하위 클래스의 입장에서는 중복적으로 가지는 속성들이 된다. 이것은 문법적으로 문제가 되지 않지만 중복되는 부분에서 속성값이 일치하지 않을 경우에 처리하는 데 대한 규칙이 없을 뿐만 아니라 의미적으로도 성립할 수 없게 된다. Fig. 8의 예제는 각 속성들이 중복되어 나타나는 것을 보여 주고 있다. Pocket1_roughing은 Tool_motion_data의 속성값으로 rough1을 가진다. 그러나 두 워킹스텝의 관련 피쳐와 사용 공구가 다르게 설정되어 있다. 이러한 경우는 작업자가 직접 작성할 때 생길 수 있는 경우를 가정한 것이다. 이런 경우에는 CNC가 NC 프로그램을 실행할 때 문제가 발생할 수 있다.

속성이 중복되어 나타나는 문제에 대해서는 CNC의 실행 모듈에게 해석상의 우선 순위를 주어 Feature_manuf_data의 하위 클래스가 가진 정보를 따르도록 한다. ISO 14649는 피쳐를 기반으로 한 NC 프로그램 작성을 목적으로 하는 만큼, 가급적 Feature_manuf_data를 중심으로 NC 프로그램을 작성하게 되며, 이때 부가적 정보로서 Generic_manuf_data를 이용하게 되기 때문이다. 그러나 이 대안은 문제에 대한 근본적인 해결책이 되지 못하며 구조적 문제점을 그대로 가지고 있어 작업자가 직접 NC 프로그램을 수정하거나 해석할 경우에는

여전히 혼돈을 가져올 수 있다. 따라서 근본적인 대안은 스키마를 수정하는 것으로서 클래스의 상속 구조와 각 클래스의 속성들을 재배치해야 한다.

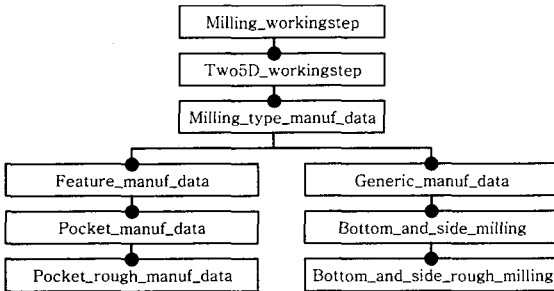


Fig. 7 Part of workingstep hierarchy

```

pocket1_rough = pocket_roughing_manuf_data(
    tool_motion_data=rough1,
    its_feature=pocket1,
    its_tool=mill_8mm,
    its_technology=techno_mill1,
    its_machine_functions=machine_fn1,
    its_id='ROUGH' );

rough1 = bottom_and_side_rough_milling(
    finishing_allowance_side=len_allowance,
    finishing_allowance_bottom=len_allowance,
    its_feature=hole1,
    its_tool=mill_12mm,
    its_technology=techno_mill1,
    its_machine_functions=machine_fn1,
    its_id='ROUGH_1' );
    
```

Fig. 8 Part of NC program with problem

6. 결론

지금까지 ISO 14649의 기능과 구조를 알아보았다. 그리고 현재의 ISO 14649는 공정 표현에 있어 유연성이 결여되는 문제점과 이미 문서화되어 있는 밀링에 대한 데이터 모델에 있어서 불필요한 중복이 발생하여 NC 프로그램 작성에서 문제가 발생할 수 있는 문제점 등을 지적하고, 이에 대한 해결 방안을 제안하였다. 이러한 문제점들은 아직 제정 단계에 있기 때문에 나타나는 문제들로서, OPTIMAL 프로젝트를 통해 어느 정도 검증은 거쳤지만, 더 많은 실제 적용 사례를 통해 검증을 거쳐야 할 것이다. 본 논문에서 제안한 비선형 공정계획과 ISO 14649를 접목하는 것과 스키마의 수

정 보완 문제도 STEP-NC와 같은 연구를 통해 ISO 14649를 지원하는 CAPP 시스템, CNC 구조 등을 구현해 볼 때, 가능성 및 새로운 대안이 제안 될 것으로 보인다. 앞으로 ISO 14649가 이러한 문제점들을 극복하고 표준으로 제정되면 ISO 6983을 대체하여 CAM과 CNC의 새로운 인터페이스로서 NC 기술의 발전 및 생산현장에 많은 기여를 할 것으로 판단된다.

Acknowledgements

The authors wish to acknowledge the partial financial support of the BK21 project.

참고문헌

1. ISO 14649, "Data model for Computerized Numerical Controllers(Draft version 10 beta)," ISO/ TC184/SC1/WG7, March 1999.
2. Djassemi, M., "A Parametric Programming Technique for Efficient CNC Machining Operations," Computers and Industrial Engineering, Vol. 35, pp. 33 - 36, 1998
3. Dhamija, D., Koonce, D. A., and Judd, R. P., "Development of a Unified Data Meta-Model for CAD-CAPP-MRP-NC Verification Integration," Computers and Industrial Engineering, Vol. 33, pp. 19 - 22, 1997.
4. Zhang, Y., Zhang, C., and Wang, H. P., "An Internet Based STEP Data Exchange Framework for Virtual Enterprise," Computers in Industry, Vol. 41, pp. 51 - 63, 2000.
5. ESPRIT III Project 8643, "Optimised Preparation of Manufacturing Information with Multi-Level CAM-CNC Coupling," ESPRIT, June 1997.