Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Robotics and Computer-Integrated Manufacturing 22 (2006) 144–153

Robotics
and
Computer-Integrated
Manufacturing

www.elsevier.com/locate/rcim

Realization of STEP-NC enabled machining

X.W. Xu*

Department of Mechanical Engineering, School of Engineering, The University of Auckland, 20 Symonds Street, Auckland, New Zealand

Received 26 May 2004; received in revised form 17 November 2004; accepted 13 February 2005

Abstract

A STEP-compliant CNC machine tool that demonstrated a G-code free machining scenario is presented. The aim of this research is to showcase the advantages of, and evaluate, STEP-NC—a new NC data model—by implementing it in a legacy CNC system. The work consists of two parts: retrofitting an existing CNC machine and the development of a STEP-compliant NC Converter called STEPcNC. The CompuCam's motion control system is used for retrofitting the machine, which is programmable using its own motion control language—6K Motion Control language and capable of interfacing with other CAPP/CAM programs through languages such as Visual Basic, Visual C++ and Delphi. STEPcNC can understand and process STEP-NC codes, and interface with the CNC controller through a Human Machine Interface. It makes use of STEP-NC information such as "Workplan", "Workingstep", machining strategy, machining features and cutting tools that is present in a STEP-NC file. Hence, the system is truly feature-based. The Application Interpreted Model of STEP-NC has been used.

Keywords: G-code; Numerical control; STEP; STEP-NC; CAD/CAM; CAPP

---

1. Introduction

Modern manufacturing enterprises are built from facilities spread around the globe, which contain equipment from hundreds of different manufacturers. Immense volume of product information must be transferred with little, if any, loss of information between the design and manufacturing departments, and even between various manufacturing facilities such as machine tools. Today's digital communication technologies have solved some critical problems of reliably transferring information across global networks. Much work though still remains to be done in the field of information exchange standards and technology. A decade ago, the description of product data has been standardized by ISO 10303 (STEP) for mechanical parts [1–3]. This leads to the possibility of using standard data throughout the entire process chain of a manufacturing enterprise. Some of the benefits have been seen in the early part of the process chain, e.g. product design and optimization, whereby the STEP output formats have been made available by most of the contemporary CAD/CAM systems. However, impediments to realizing this principle in its entirety are believed to be the data formats used at the machine tool level. Most Computer Numerically Controlled (CNC) machines are programmed in the ISO 6983 "G-code" language [4], a standard developed and effectively implemented by many CNC machines back in the 1950s through to 1970s. The background of the CNC equipment at that time is mush like that of the computer systems in that era where punched cards and tapes were the main information medium and the computing power was only a tiny fraction of what the current computers are offering. The interface between Computer-Aided Process Planning (CAPP) system and a CNC machine tool is through programs that are generated by functions in a CAPP or Computer-Aided Manufacturing (CAM) system, typically known as "post-processors". Such an

---

*Tel.: +64 9 373 7599x84527; fax: +64 9 373 7479.
E-mail address: x.xu@auckland.ac.nz.

0736-5845/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2005.02.009

interface has long caused bottleneck problems for the manufacturing industry, in particular limited the program portability and that of a CNC machine. First of all, the language focuses on programming the cutter's centre location path (hence the name CL data file) with respect to the machine axes, rather than the machining process/tasks with respect to the part. Secondly, the standard defines the syntax of program statements, but in most cases leaves the semantics ambiguous. Thirdly, vendors usually supplement the language with extensions that are not covered in the limited scope of ISO 6983, only to have made the CNC programs more proprietary and hardware dependent.

ISO 14649 [5], otherwise known as STEP-NC, is a new model of data transfer between CAD/CAM systems and CNC machines. It remedies the shortcomings of ISO 6983 by specifying machining processes rather than machine tool motion, using the object-oriented concept of "Workingsteps". Workingsteps correspond to high-level machining features and associated process parameters. As STEP-NC is an extension of STEP to handling NC processes, it strictly follows the STEP standard [1]. In a STEP-NC file, for example (Fig. 1), the HEADER contains general information and comments about the part program, such as filename, author, date and organization. The DATA section is the main section of the program, containing all the information about manufacturing tasks and geometries. The data are divided in three significant parts, (a) workplan and executables; (b) technology description and (c) geometry description. CNC machines are therefore provided with rich information for part manufacture. CNC machine tools are also made responsible for translating information defined by STEP-NC such as Workingsteps to axis motion and tool operation. Another major benefit of STEP-NC is its compliance with ISO 10303, offering an attractive solution to total integration of CAD/CAPP/CAM/CNC without information exchange barriers [6]. Effectively, the Application Interpreted Model (AIM) for ISO 14649, which is an Application Reference Model (ARM), in STEP terminology, is being developed and will become a STEP Application Protocol AP238 [7].

A gradual evolution from ISO 6983 programming to portable, STEP-compliant feature-based programming has been envisioned [9]. Early adopters of STEP-NC will need to support data input of legacy G-code manually or through programs, just as modern controllers support both command-line interfaces and graphical user interfaces. Prototypes of this kind of STEP-NC work have been demonstrated by both European and American research groups [6,10]. The next milestone for STEP-NC enabled CNC is to use a new CNC controller which is not driven by G-code but by STEP-NC files directly or indirectly.

The research work reported herein is on the development of a CNC controller that is G-code free and capable of processing STEP-NC information. There are two pieces of work in this research, retrofitting a legacy CNC lathe and developing a STEP-NC Converter–STEPcNC Converter.

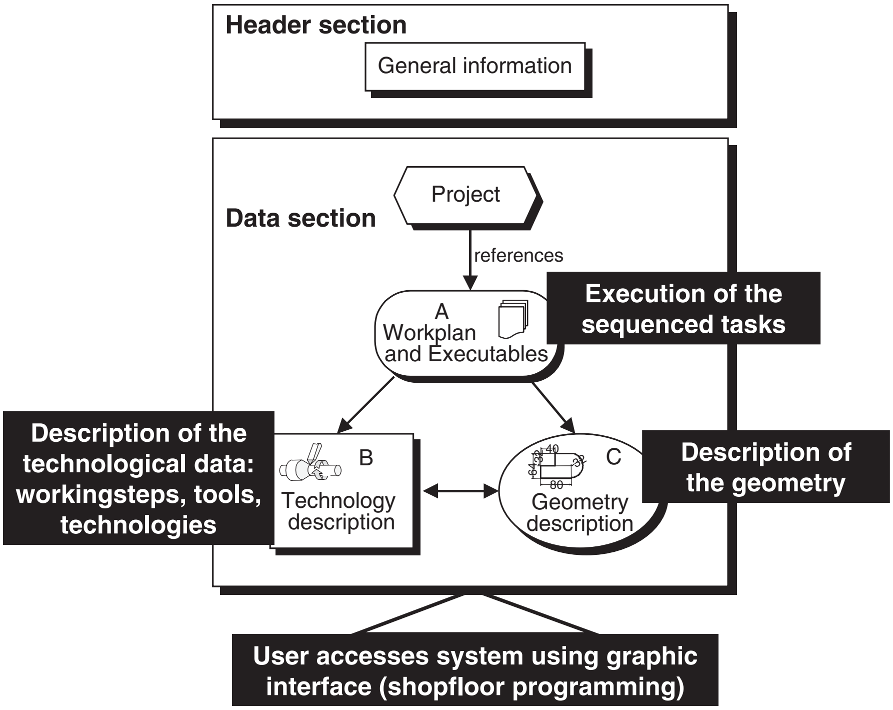


Fig. 1. Data structure of a STEP-NC file [8].

2. Literature review

Substantial research work has been carried out by research groups from countries such as Germany, Switzerland, the UK, Korea, the US and New Zealand. Some of the projects have been coordinated across a few countries and others are not. The remaining section only describes the related research work.

A prototype system named STEPturn has been developed at ISW Stuttgart [11]. It adopts STEP and STEP-NC standards for turned parts. Generally speaking, STEPturn is a CAPP system bridging CAD and CAM. STEPturn first reads geometry data from a STEP AP-203 Part 21 file, and then performs normal process-planning tasks such as feature recognition and Workingstep sequencing in order to generate a STEP-NC physical file, that is an ISO 14649 file. The Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University focuses on STEP-compliant CAM and NC controllers. It has developed the first industrial prototype of a STEP-compliant NC controller based on the Siemens 840D controller [10,12]. A graphical user interface has been developed, using Shop Mill (a shop floor oriented NC programming tool) and Sinumerik 840D HMI (Human Machine Interface). This interface, backed up by a so-called STEP-NC Parser Library [10], can process STEP-NC program files. The Workingsteps are executed based on the existing NC-cycles, or directly compiled to yield switching commands and processed to control specific geometrical interpolation formats. An Agent-Based CAM system (AB-CAM) has been developed at the Advanced Manufacturing Sciences and Technology Centre of Loughborough University [13–16]. It is a system with a STEP compliant NC data structure. The STEP compliant process plan generator in the system can select a process type, machine tool, cutting tool, cutting parameters, fixture methods and clamping locations to transfer a feature in AP224 format into a Workingstep in a STEP-NC file. The AB-CAM system does not have an intermediate stage. The output from the AB-CAM system is native STEP-NC files.

The Korean STEP-NC system has been developed at NRL-SNT (National Research Laboratory for STEP-NC Technology) in Pohang University of Science and Technology [17–20]. In preparation for executing a STEP-NC program, two different approaches have been used, a STEP-NC enabled CNC controller and a conventional CNC controller. In the STEP-NC enabled CNC controller, NCK/PLC has been developed and used to convert the STEP-NC data model into machine tool motion. It interfaces with machine tool hardware (drivers and motors) via an I/O board. If a conventional CNC (only accepting ISO 6983 code) has to be used, conversion from ISO 14649 to ISO 6983 is necessary. This requires explicit tool path computation in which the cutting tools are used should be the same as specified in the ISO 14649 part program.

In the NIST Intelligent Systems Division (ISD), an ISO14649 (ARM) Interpreter has been developed. The interpreter reads a STEP Part 21 file based on the schemas in Parts 10 and 11 of ISO 14649 and generates canonical machining commands, which are used by the Enhanced Machine Controller (EMC) developed at ISD. ISD is now developing an AP238 (AIM) Interpreter.

3. Proposed research

The system being developed in the Manufacturing Systems Group at the University of Auckland has three objectives: to make product data interchangeable, product information flow seamless and the system itself independent of any CAD/CAM systems. STEP-NC is adopted as the “common language” for NC manufacturing as is STEP for the design processes. Fig. 2 depicts the information flow in the system which is accompanied by different “derivatives” of STEP standards at different stages, hence both product data interchangeability and seamless product information flow can be achieved.

The first move toward “common language”-based product development is to work with design models in

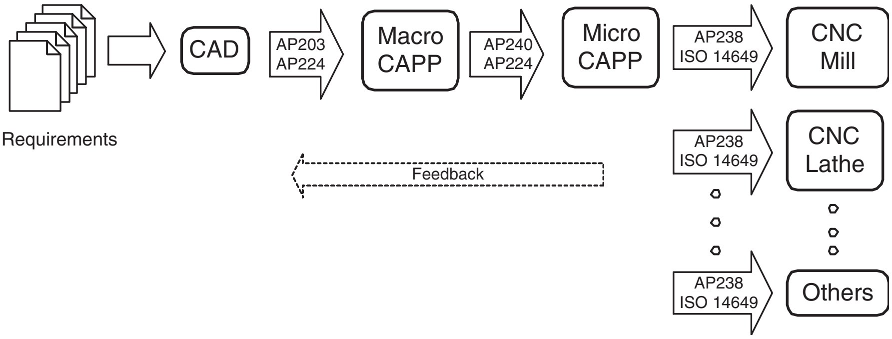


Fig. 2. Integrating design with manufacturing via STEP/STEP-NC.

STEP (AP 203) format. A product model defined in terms of pure STEP geometry is re-interpreted in terms of the manufacturing features stipulated by STEP AP224 [21]. Then, macro process planning is carried out based on these manufacturing features. The outcome of this macroprocess planning is a ISO 10303 AP240 [22] file, and the microprocess planning yields a ISO 10303 AP238 file, i.e. an STEP-NC AIM file. This file includes the information about the design model itself, the stock, its manufacturing features, the tool/fixture requirements, the manufacturing process sequence and etc. It is self-documenting and allows complete safety checking. This type of information can be described as "what-to-do" information, meaning only manufacturing tasks are described, but not the way(s) of completing them. This type of "what-to-do" information is then fed to a STEP-NC controller to generate so-called "how-to-do" information, which may still be documented in an ISO 14649 file, i.e. an ARM file. This type of information then becomes machine-specific and is used to issue low-level NC commands to drive the machine tool. Following the design-manufacturing process chain illustrated in Fig. 2, the system consists of four main modules (Fig. 3), (A1) Generation of Generic STEP-NC Programs; (A2) Generation of Native STEP-NC Programs; (A3) Generation of Native CNC Commands and (A4) Execution of the Process Plan on a CNC Machine.

This paper discusses the work being carried out corresponding to A3 and A4, i.e. interpret the STEP-NC information in this AIM format, generate the low-level machine control commands and execute them on the machine tool. Interested readers can refer to previous publications [23–25] for details on other parts of the system. It is worth noting that the low-level machine commands are suppressed, only the STEP-NC is visible to a user. The user has full control over and access to the process plan represented in the STEP-NC format through the interpreter interface. Once the process plan is confirmed, execution of the program is done automatically and autonomously by the controller software. The legacy machine tool is a training lathe, HERCUS PC200. The work can be divided as two distinctive parts, retrofitting the machine tool and developing STEPcNC.

4. Retrofitting a CNC Lathe

Interfacing with STEP-NC information needs a more open and robust CNC controller. Like all the other current CNC controllers, the existing controller on the HERCUS PC200 can only handle G-code commands for operations. Its interface to outside is limited and not user-friendly. Therefore, retrofitting the lathe is a must for deploying the STEP-NC concept.

4.1. The motion control system

The 6K motion control system from Compumotor [26] has been opted to retrofit the lathe. There are a number of reasons for such a decision. The 6K offers a re-configurable multi-axis control (2–8 axes), servo or stepper interchangeability, expandable I/O, multitasking and choices of communication means.

With 6K motion control system, both servo and stepper type of controls can be easily catered for. Extendable digital and analogue I/O modules provide the necessary flexibility for configuring the machine tool. Another main reason of selecting the 6K system is that it

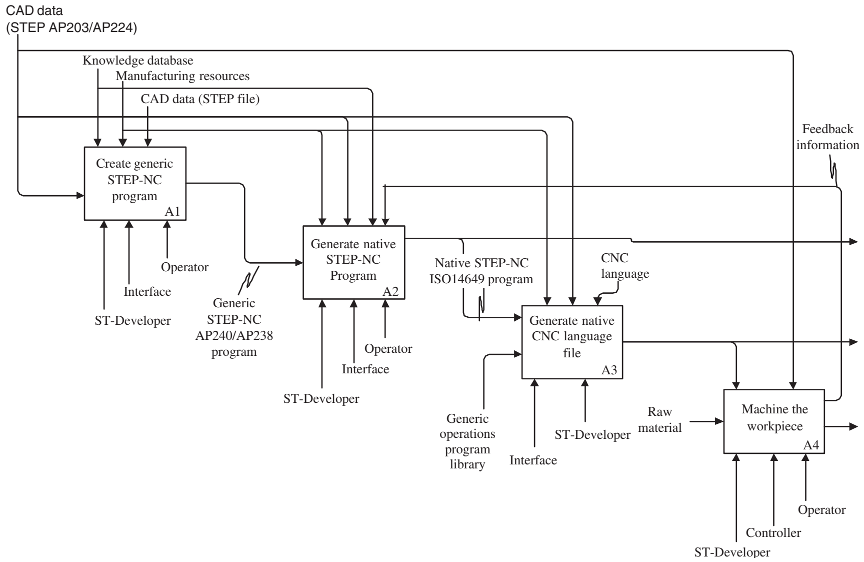


Fig. 3. IDEF0 diagram of the STEP-compliant CAD/CAPP/CAM/CNC system.

is easy to accomplish multiple independent control sequences as they are so often seen on a machine tool. This is accomplished by the Motion Planner which comes with the system and can interface easily with the 6K Controller in the Windows environment. Applications can be developed using Visual Basic, Visual C++ or Delphi. Utilizing object-oriented wizards, a user can easily set up a controller, write a program, check syntax, download and run programs and tune axes. Communications can be made in three different ways, RS 232 serial ports, Bus-based control and Ethernet (LANS or Internet).

4.2. Retrofitting

4.2.1. Hardware connections

The HERCUS PC200 lathe is a typical two-axis lathe with a tool turret that can hold eight cutters. Both the chuck and lid are pneumatically driven. Table 1 summarizes the components that have been retrofitted and their functions.

A number of hardware components have been used for retrofitting. These components include a Compumotor 6K4 Controller, two servo motors and their drivers for *X*- and *Z*-axes, an OEM 1000 watt power transformer, an EVM32 (expansion input/output module) and a VM25 (25 pin extension) unit.

To consolidate the controller, only VM25 is used alongside a microcontroller powered at a maximum speed of 20 MHz. It converts VM25's four digital outputs into 16 digital outputs, though only 10 are required for the current retrofitting job. This is done by treating each of the different combinations of the four outputs as single digital signals. Fig. 4 shows a flow diagram detailing the physical connections as well as information flow of the retrofitted lathe.

The 6K4 controller sends and receives feedback/commands from the computer and transfers the digital signals through VM25 to and from the limit switches and the microcontroller. The limit switches are used to home the *X*- and *Z*-axis motors. The microcontroller controls various lathe components such as tool turret, lid and chuck. The spindle driver is controlled by a 0–10 V analogue signal from 6K4, which provides the spindle motor with a 0–120 V signal. The spindle encoder sends the speed signal back to the controller and also to an external display panel. To control the *X*- and *Z*-axes, the controller sends 0 to ±10 V analogue signals to the *X*- and *Z*-axis drivers which provide the motors with a 0 to ±70 V signal. The motors also provide Hall effect (positional) feedbacks to the drivers, and the encoders provide the motor speed feedback signals to the controller.

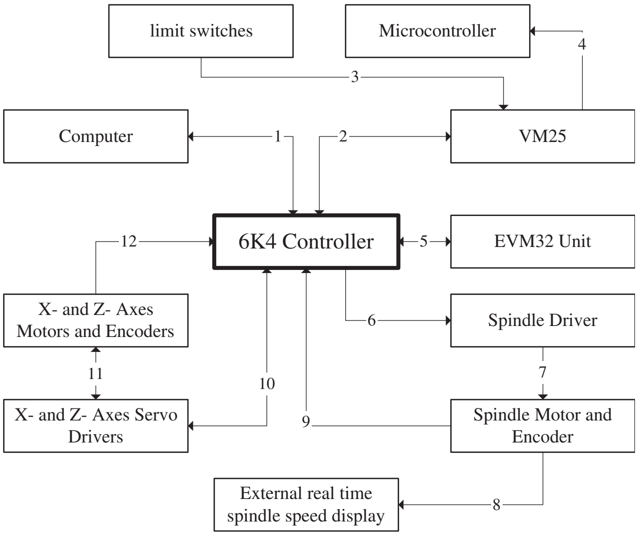


Fig. 4. Connections within the retrofitted lathe.

Table 1
Retrofitted components and their functions

Components	Functions
X- and *Z*-axis motors	Speed control, bi-directional
Tool turret	Index, home and lock in position
Spindle motor	Speed control, anticlockwise
Chuck	Grip or release
Lid	Open or close
Lubricant	On or off

4.2.2. System tuning and set-up program

System tuning is through Compumotor's Motion Planner software and a set-up program is created to forego any application programs. The Motion Planner can download a program and run it in real time. Compumotor Motion Planner is a software package specifically designed for controlling Compumotor's controllers [26]. The software package utilizes 6K language, a basic ASCII language that contains commands to control hardware devices linked to the 6K controller. The 6K language is capable of sending two types of commands, analogue and digital. The analog commands are used to set different parameters of a machining operation with real numbers, e.g. parameters of distance, velocity and acceleration, whereas the digital commands use Boolean operators to set the state of a component, e.g. the state of a drive (Enable/Disable) and the state of an output (On/Off). The Set-up program, once completed, contains all the features which determine the characteristics of the machine. Four sets of parameters are of particular importance in a Set-up program. They are Drive Set-up, Scaling Set-up, Encoder Set-up and Limits/Home.

4.2.3. Machining commands

In order to ease the job of linking STEP-NC information with the retrofitted machine tool, the turning operations defined by STEP-NC (ISO/DIS 14649 12 [27] and ISO/DIS 14649 121 [28]) are used to develop the machining commands.

(1) Turning operations defined by ISO/DIS 14649–12

ISO/DIS 14649 12 defines a base class for all turning operations, which contains information about "technology" and cutting "strategy". It is a subtype of *machining_operation* entity defined in ISO 14649–10 [29]. There are two basic categories of machining operations: roughing and finishing; each can be one of the following six operations: facing, grooving, cutting_in, contouring, threading and knurling. The following EXPRESS schema describes a generic turning operation,

```
ENTITY turning_machining_operation
    ABSTRACT SUPERTYPE OF (ONEOF(facing, grooving, cutting_in, contouring, threading, knurling))
    SUBTYPE OF (machining_operation);
    approach : OPTIONAL approach_retract_strategy;
    retract : OPTIONAL approach_retract_strategy;
    its_machining_strategy : OPTIONAL turning_machining_strategy;
END_ENTITY;
```

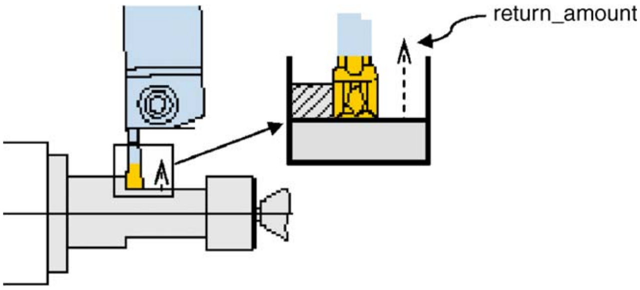


Fig. 5. Grooving operation.

The "approach" strategy is optional. If multiple layers have to be cut, this strategy will be used to move from one layer to the start point of the next. Also optional is the "retract" strategy after finishing the last cut. By default, the NC controller determines the retract strategy. It may choose not to use any retract movement if the end point of cutting coincides with the start point of next cutting operation. The "its_machining_strategy" attribute describes the detailed strategy for executing a turning operation, i.e. creating a turning toolpath. It is a subtype of entity *machining_strategy* defined in ISO 14649–10.

(2) The Grooving Operation—an example

To illustrate, the development of the 6K grooving program has been described here. Grooving, as defined by STEP-NC, is a plunging operation in the direction perpendicular to the component axis (Fig. 5) [27]. For machining the side or wall of a groove, other operations such as *facing* and *contouring* are used. The EXPRESS schema for grooving is given below.

```
ENTITY grooving
    ABSTRACT SUPERTYPE OF (ONEOF(grooving_rough, grooving_finish))
    SUBTYPE OF (turning_machining_operation);
    return_amount : OPTIONAL length_measure;
    dwell_time_bottom: OPTIONAL time_measure;
    allowance : OPTIONAL length_measure;
END_ENTITY;
```

"return_amount" is the amount of path return for cutting the next layer (Fig. 5). "dwell_time_bottom" is the time the cutter dwells at the bottom of the groove. "Allowance" is a depth of material to be left over for a

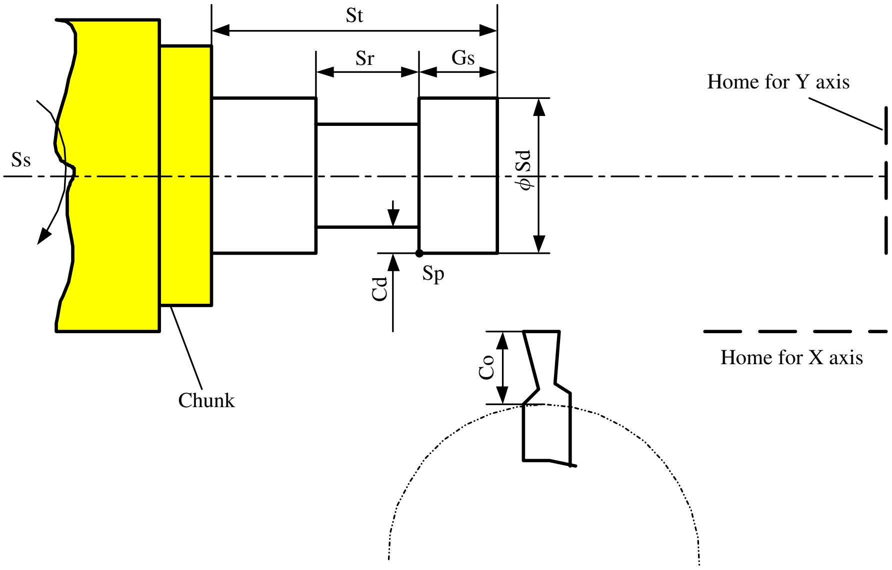


Fig. 6. Parameters for grooving cut.

finishing operation. In order to generate a complete set of information for a grooving cut, other information such as the geometry of the groove feature and the workpiece needs to be utilized. The "groove" feature is a subtype of "revolved_feature" formed by sweeping an "open_profile". "revolved_feature" is one of the subtypes of "turning_features",

```
ENTITY groove
  SUBTYPE OF (revolved_feature);
  sweep : open_profile;
END_ENTITY;
```

"Turning_features" are referenced from AP224 of ISO10303; i.e., fully compliant with ISO10303-224. These features are taken as standard for STEP-NC data model, for the sake of compliance with the existing standard ISO10303-224. The reference path for a groove feature is as follows:

```
revolved_profile < =
  {feature_definition = > instanced_feature}
feature_definition < = characterized_object
  {characterized_object
    characterized_object.description = 'groove'}
```

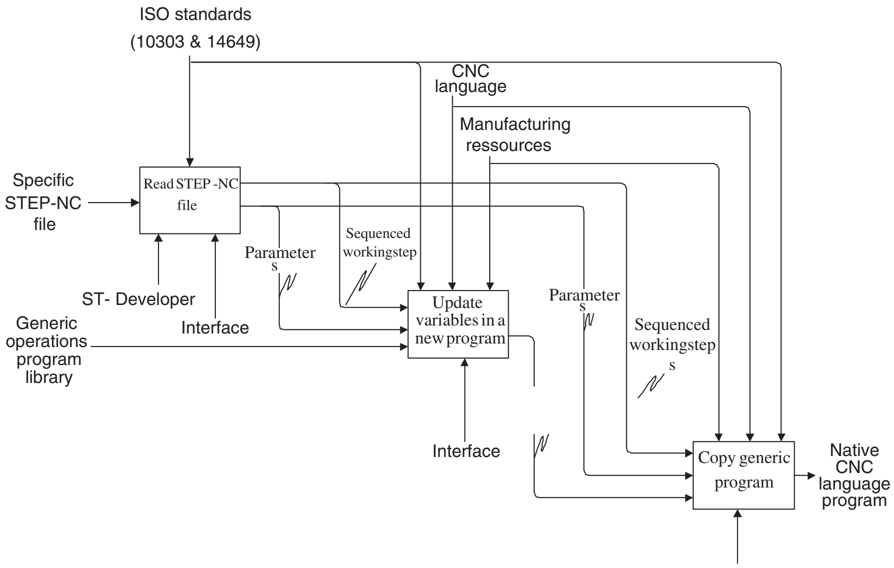


Fig. 7. Generate a native CNC program.

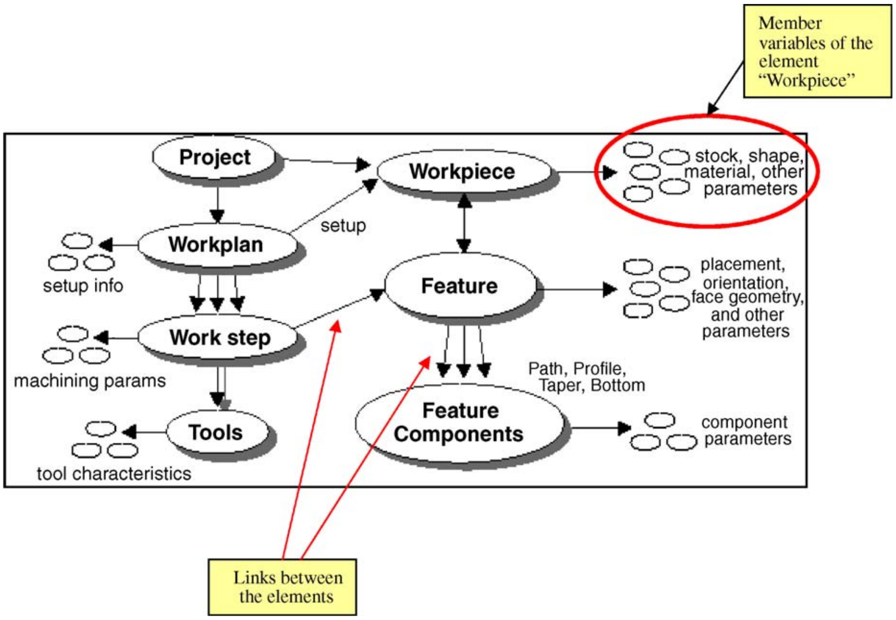


Fig. 8. Structured information in a STEP-NC file.

For the retrofitted lathe to machine a groove, the following set of parameters are extracted and used to create a 6K program (Fig. 6):

- stock length (*St*);
- stock diameter (*Sd*);
- stock removal (*Sr*);
- cut depth (*Cd*);
- spindle speed (*Ss*);
- groove starting point (*Sp*);
- groove starting length (*Gs*);
- cutter offset (*Co*).

Using these parameters, the cutter path is generated and a program called GROOVE.prg is created, tested and "packaged" so that the program can be used as a subroutine with only seven parameters to be specified before execution. Similar subroutines have been developed for other Workingsteps defined in STEP-NC. Therefore, the system effectively contains a library of Workingstep-based 6K function templates. With this library, mapping STEP-NC information to 6K commands is made easier. The only work remaining is to extract the required information from a STEP-NC file and populate the targeted 6K template for final execution.

5. STEPcNC Converter

The STEPcNC Converter serves as an interface between a STEP-NC file and the CNC lathe. It is programmed using Microsoft Visual C++ for an automatic translation of a STEP-NC program into 6K language codes—a native CNC language program (Fig. 7), which can drive the retrofitted lathe.

STEP Tools STIX [30] has been used to extract the required information. STIX is a STEP IndeX library for STEP AP-238. It contains a C++ library which provides useful functions to process manufacturing data in STEP AP-238 format. Therefore, STIX simplifies implementation and processing of STEP AP-238 information in programs written in Microsoft Visual C++. STIX uses so-called "cursers" to navigate through an ordered structure of STEP-NC entity (Fig. 8). At the present time, mapping ISO 14649–12 and ISO 14649–121 to AP238 is not yet completed. Accessing turning information is limited with the current STIX. Therefore, STEPcNC processing capability is only partial.

There are two files generated, a 6K program file containing the machining commands and a report file presenting a readable summary of the machining operations and errors if any. Fig. 9 shows the conversion process.

A 6K program always starts with the initialization information, such as homing the axes, re-setting the variables, creating a name for the operation, adding a date and time. For each Workingstep, actions such as

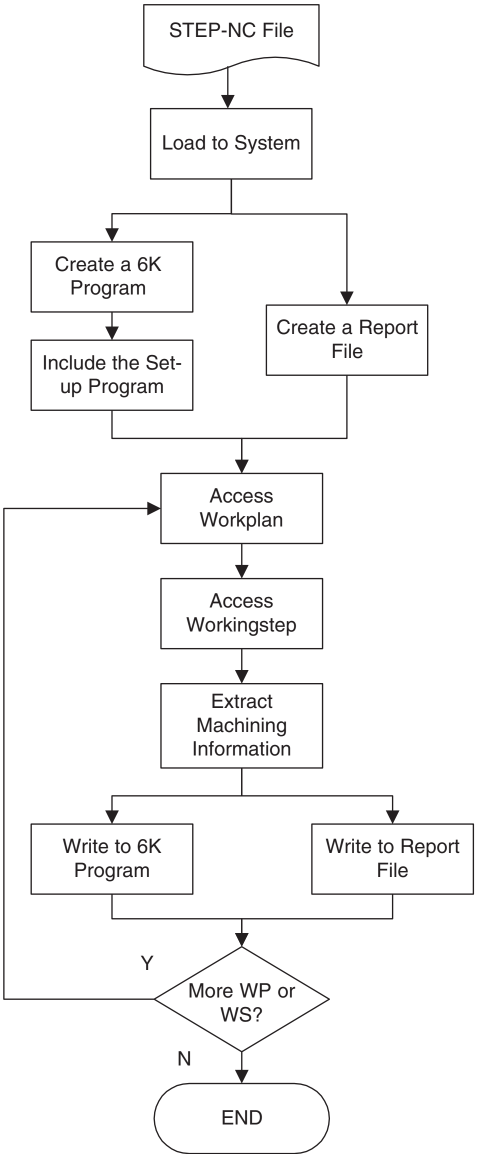


Fig. 9. Creation of a 6K program and report file.

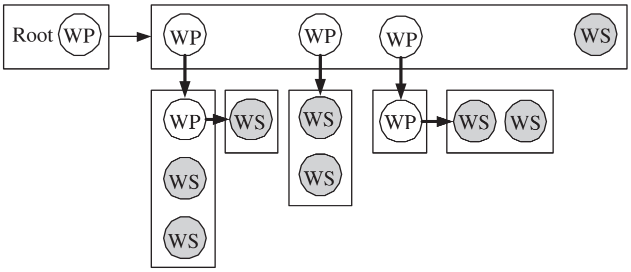
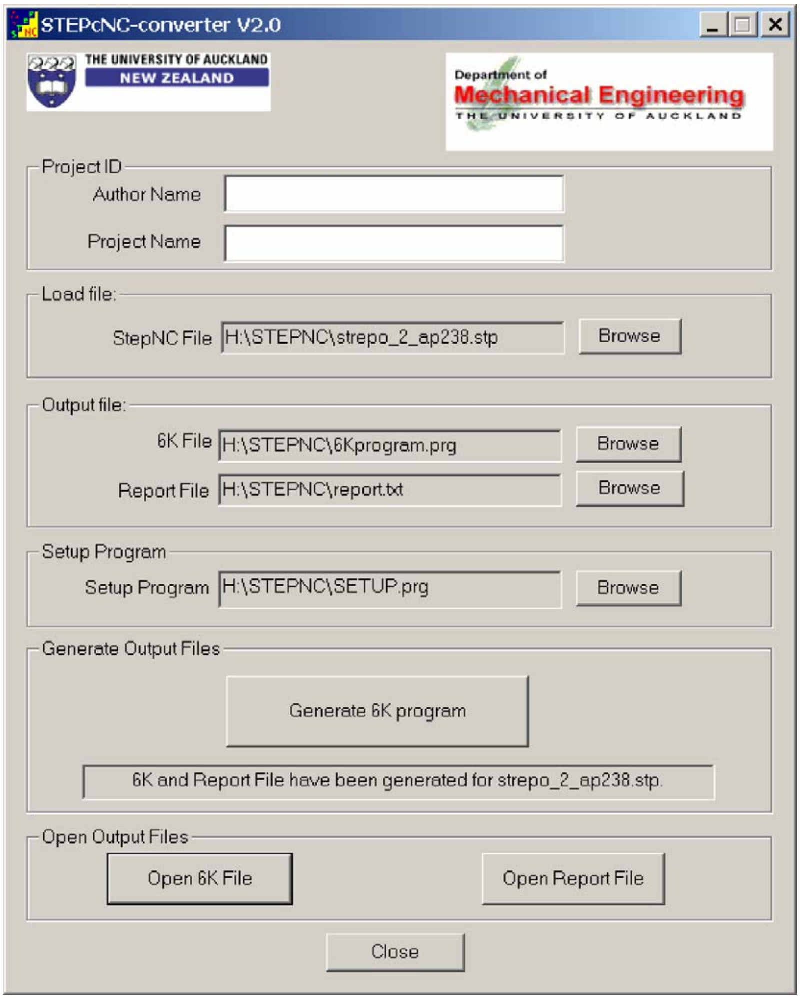


Fig. 10. Navigating through a STEP-NC file.



StepNC File: H:\STEPNC\strepo_2_ap238.stp

6K File: H:\STEPNC\6Kprogram.prg

Report File: H:\STEPNC\report.txt

Setup Program: H:\STEPNC\SETUP.prg

6K and Report File have been generated for strepo_2_ap238.stp.

Fig. 11. STEPcNC Converter interface.

loading a cutter, turning on coolant, and closing the lid are saved into the 6K program. Afterwards, machining parameters such as cutting velocity, federate and depth of cut are written to the program. Since a Workplan may include both Workplan(s) (WP) and Workingstep(s) (WS) (Fig. 10), a set of self-looping codes are used to traverse through all the Workplans and Workingsteps.

Fig. 11 shows the interface of STEPcNC. In addition to generating a 6K program and a report file, the interface can also load an existing 6K program or report file to browse and/or modify. This is done by evoking a Motion Planner editing window through the OEM feature in MFC.

6. Conclusions

Use of STEP standards throughout the design-manufacturing process chain enables and simplifies data exchange. The retrofitted CNC lathe has a more open and flexible architecture, making it easier to integrate with other manufacturing facilities, e.g. workpiece handling device. The developed STEPcNC Converter alongside the retrofitted CNC lathe realized a G-code free machining scenario. As the retrofitted machine tool can understand STEP-NC information, the product design and manufacturing information, once represented using different STEP Application Protocols as shown in Fig. 2, can be exchanged with little effort, hence realization of the first and the second objectives set up in this research. Complete design information is passed on, and made available at the machine side, making editing machining information an easy task. Availability of design information at the machining stage also enables a reliable collision check, accurate simulation and feedback from the machining stage to the design stage. Since the design and manufacturing information coded in a STEP-NC file has been kept machine tool independent, its portability is guaranteed. It is necessary though that different STEP-NC interpreters be used for the machine tools with a different NC kernel. Admittedly, 6K has no further advantage over G-code in its role as a control language between

STEP-NC and the low level motion control commands. It is nonetheless more open and easier to program and interface with other applications. It is the motion control system programmed and driven by the 6K language that offered far more functionalities to the machine tool itself.

Acknowledgment

The author wishes to acknowledge the contributions made by a number of research students including, in the chronological order of their involvement in the project, Janak Patel, Arun Patel, Renaud Gardes, Arnaud Gage, Christopher Ng, Chia Sin Lee, Michel Wagner, Mao Jin, Wang Hongqiang, Ziad Burjony and Nicolas Giraldon.

References

[1] International Standards Organization, ISO 10303-1. Industrial automation systems and integration—product data representation and exchange—part 1: overview and fundamental principles. 1994.

[2] International Standards Organization, ISO 10303-203. Industrial automation systems and integration—product data and exchange—part 203: application protocol: configuration controlled 3D designs of mechanical parts and assemblies. 1994.

[3] International Standards Organization, ISO 10303-21. Industrial automation systems and integration—product data representation and exchange—part 21: implementation methods: clear text encoding of the exchange structure. 2002.

[4] International Standards Organization, ISO 6983/1. Numerical control of machines—program format and definition of address words—part 1: data format for positioning, line and contouring control systems, 1st ed. 1982.

[5] International Standards Organization, ISO 14649-1. Data model for computerized numerical controllers: part 1—overview and fundamental principles. 2003.

[6] Xu X, He Q. Striving for a total integration of CAD, CAPP, CAM and CNC. Robot Comput-Integr Manuf 2004;20:101–9.

[7] International Standards Organization, ISO/DIS 10303-238. Industrial automation systems and integration—product data representation and exchange—part 238: application protocols: application interpreted model for computerized numerical controllers. 2003.

[8] STEP-NC Newsletter. Issue 3, November 2000, http://www.stepnc.org/.

[9] Hardwick M. Digital manufacturing using STEP-NC, AUTOTECH 2001 conference and exhibition, AIAG, August 28–30, 2001. USA: Detroit's Cobo Centre.

[10] Weck M, Wolf J. ISO 14649 provides information for sophisticated and flexible numerically controlled production, Prod Eng [WGP-Annals] 2002.

[11] Storr A, Heusinger S. STEP-NC–Grundlage einer CAD/NC-Prozesskette Das STEP-NC-Prozessmodell für die Drehbearbeitung. 2002.

[12] Weck M, Wolf J, Kiritsis D. STEP-NC–The STEP compliant NC programming interface evaluation and improvement of the modern interface. Proceedings of the ISM project forum, Genf, October, 2001, Monte Verita/Ascona Schweiz.

[13] Allen Jr RD, Rosso RSU, AB-CAM: An agent-based methodology for the manufacture of STEP compliant feature based components.

[14] Allen RD, Newman ST, Harding JA, Rosso Jr, RSU. The design of a STEP-NC compliant agent based CAD/CAM system, Proceedings of the 13th international conference on flexible automation and intelligent manufacturing conference (FAIM2003), Tampa, Florida, USA. pp. 530–540.

[15] Rosso Jr. RSU, Allen RD, Newman ST. Future issues for CAD/CAM and intelligent CNC manufacture, 19th international manufacturing conference—IMC-19. Belfast: Queen's University. 2002.

[16] Newman ST, Allen RD. CAD/CAM solutions for STEP-compliant CNC manufacture. Int J Comput-Integr Manuf 2003; 16(7–8):590–7.

[17] Suh SH, Cheon SU. A framework for an intelligent CNC and data model. Adv Manuf Tech 2002;19:727–35.

[18] Suh SH, Cho JH, Hong HD. On the architecture of intelligent STEP-compliant CNC. Int J Comput-Integr Manuf 2002;15(2): 168–77.

[19] Suh SH, Chung DH, Lee BE, Cho JH, Cheon SU, Hong HD, Lee HS. Developing an integrated STEP-compliant CNC prototype. J Manuf Syst 2002;21(5):350–62.

[20] Suh SH, Lee BE, Chung DH, Cheon SU. Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. Comput-Aid Des 2003;35(12):1069–83.

[21] International Standards Organization, ISO 10303-224. Industrial automation systems and integration—product data representation and exchange—part 224: application protocol: mechanical product definition for process plans using machining features. 2001.

[22] International Standards Organization, ISO/DIS 10303-240. Industrial automation systems and integration—product data representation and exchange—part 240: application protocol: process plans for machined products. 2004E.

[23] Wang H, Xu X. A STEP-compliant 'adaptor' for linking CAPP with CNC. 34th international MATADOR conference, 7–9 July. Manchester, UK: UMIST; 2004.

[24] Xu X, Mao J. Development of the client tier for A STEP compliant CAPP system. Proceedings of the sixth international conference on frontiers of design and manufacturing (ICFDM'2004), 21–23 June 2004, Xi'an, China.

[25] Xu X, Mao J. A STEP-Compliant collaborative product development system. Proceedings of the 33rd international conference on computers and industrial engineering, March 25–27, 2004, Ramada Plaza···Oriental Hotel, Jeju, Korea, CIE598.

[26] http://www.compumotor.com.

[27] International Standards Organization, ISO 14649-12. Data model for computerized numerical controllers: part 12: process data for turning. 2003.

[28] International Standards Organization, ISO 14649-121. Data model for computerized numerical controllers: part 121—tools for turning. 2003.

[29] ISO 14649-10. Data model for computerized numerical controllers: part 10—general process data. 2003.

[30] http://www.steptools.com/stix.