INT. J. COMPUTER INTEGRATED MANUFACTURING, 2002, VOL. 15, NO. 2, 168–177

Taylor & Francis
Taylor & Francis Group

On the architecture of intelligent STEP-compliant CNC

SUK-HWAN SUH, JUNG-HOON CHO and HEE-DONG HONG

Abstract. STEP-NC or ISO 14649 is a new data model between CAM and CNC, currently under development by ISO TC184 SC1 and 4. Distinguished from the present CNC language (ISO 6983), the new data model includes rich information enabling feature-based programming and control for manufacturing operation. This paper aims to present a conceptual framework for an intelligent STEP-compliant CNC (called ASNC: Autonomous STEP-compliant CNC) taking ISO 14649 as an input and carrying out manufacturing tasks in an intelligent and autonomous manner. The framework is derived from the analysis of information contents of ISO 14649, and the role of CNC on the shop floor of an intelligent manufacturing system. To show the mechanism of ASNC, an operational scenario with functional details for NC milling operation is presented. The scenario illustrates that the new CNC can autonomously carry out the manufacturing tasks based on ISO 14649 and can deal with unexpected changes on the shop floor.

1. Introduction

Current NC (numerical control) programming is based on ISO 6983, called G-code (International Standards Organization 1982), where the cutter motion is mainly specified in terms of position and the feedrate of axes. Since the G-code is a low level language, it is hard for the shop floor operator to calculate the process flow, especially as the part program gets complicated. Furthermore, because G-codes deliver only limited information to CNC (computer numerical control), excluding such information as part geometry and process plan, it has posed serious problems, making CNC little more than an executing mechanism completely unaware of the motions being executed. Therefore, for intelligent control, CNC should be capable of accommodating information on the machining task, including the product model and process plan.

Authors: National Research Laboratory for STEP-NC Technology (NRL-SNT), Pohang University of Science and Technology, San 31, Hyoja-dong, Pohang, 790-784, Korea. e-mail: shs@postech.ac.kr

Recently, a new and comprehensive language for CNC has been under development by ISO TC184 SC1 and SC4 (International Standards Organization 2000). The new language, formalized as ISO 14649, is based on the product model STEP (ISO 10303) together with process plan information. As described in section 2, ISO 14649 specifies information contents and semantics (ICS) for various CNC manufacturing processes. It is currently available in the DIS-ballot version (International Standards Organization, 2000), with its final version to be completed in a few years. Upon completion, it will be a new CNC language replacing ISO 6983.

As the new language is being established, increasing attention is being paid to the development of new CNCs based on the new interface scheme between CAM and CNC. Among these, two — *Super Model* of STEP Tools and *NC Prototype* of Europe — were reported in the recent ISO TC 184 SC4 meeting in October 2000 (Hardwick 2000). Super Model places an emphasis on the development of an 'intelligence interface' between the ISO 14649 database and CNC via XML, while NC Prototype incorporates the ISO 14649 interpreter in existing CNC. Despite some differences in these prototype systems, the scope of their target system is, in essence, a 'basic type' of CNC, where CNC faithfully follows as specified in the part program given by the ISO 14649 scheme.

Although the primary purpose of ISO 14649 is to establish the new interface standard, another important aspect; i.e. providing CNC with rich information for intelligent control, should not be underestimated. In general, the information contents of ISO 14649 include various information for CNC to control the manufacturing operation intelligently and autonomously, but on limited terms. To enhance the intelligence and autonomy of CNC, ISO 14649 needs to be updated in some parts, as pointed out in Suh (2001). In other

International Journal of Computer Integrated Manufacturing
ISSN 0951-192X print/ISSN 1362-3052 online
http://www.tandf.co.uk/journals
DOI: 10.1080/09511920110056541

words, since the capability of the STEP-compliant CNC is largely dependent on how valuable information is utilized within CNC, it is critical to develop an architecture that can take full advantage of the new language containing rich information.

As far as NC architecture is concerned, several schemes have been proposed. Holonic numerical controller (HNC) (Kruth 1994, Suh *et al.* 1999) is based on the paradigm of a holonic manufacturing system (HMS) (HMSFPD 1995, Brussel *et al.* 1997), where the NC holon is characterized by autonomy and cooperation. TRUE-CNC (Yamajaki *et al.* 1997) is a comprehensive CNC system with CAD, CAPP, CAM, CNC, monitoring and inspection functions. However, the STEP-based new language (ISO 14649) has not been taken into consideration in the architecture, except in the one given by Brouer and Weck (1997).

The industrial aspect of CNC architecture is related to the implementation methodology. Most CNCs developed up to the present time are of a 'closed' architecture, without allowing the users to access the internal functions for any modification. Developing an 'open' CNC architecture has been one of the main concerns for both CNC developers and users since the mid 1980s. However, it was not until the early 1990s that a few commercial products were investigated by some leaders of the CNC industry. Liu and Yamajaki (1988) emphasized the open architecture to realize agile production and autonomous CNC control. However, a truly open CNC is still under development with different architectural schemes, such as OSACA of Europe (ESPRIT 1995), OMAC of USA (Chrysler *et al.* 1994), and OSEC of Japan (OSEC Consortium 1998).

In this paper, we propose a conceptual framework, together with an implementation strategy for an intelligent STEP-compliant CNC (called 'ASNC': Autonomous STEP-compliant CNC), taking ISO 14649 as the input and controlling the machining task autonomously. 'Autonomous' here means that the CNC is able to plan the machining sequence, generate the tool path, execute the generated tool path, and to monitor the execution status. It is thus distinguished from the 'basic type' mentioned above, where CNC faithfully executes as programmed without an autonomous capability. The ASNC presented in this paper is based on the analysis of information contents of ISO 14649, and the role of CNC on the shop floor in an intelligent manufacturing system. It is a picture for the next generation CNC, which can be realized in the near future.

In addition, pointing out the limitation of the current scheme (specifically, Part 10 of ISO 14649 DIS version) in representing a nonlinear process sequence, we proposed a new scheme based on the concept of 'process sequence graph'. Through an example, we confirmed the necessity of a new scheme to make CNC intelligent and autonomous.

In section 2, a brief overview on the information contents of ISO 14649 and its interface scheme with CNC are provided, followed by the description of some crucial considerations to be taken in designing the intelligent STEP-compliant CNC in section 3. Then, an architecture of ASNC, together with an operational scenario including a new proposition (process sequence graph) is presented in section 4, with a few concluding remarks given in section 5.

2. ISO 14649 and interface scheme

As mentioned earlier, the details of ISO 14649 are still under revision, indicating possible changes in it. However, it is expected that the basic structure and information contents of ISO 14649 will be maintained in the new version. In this section, we provide an overview on the information contents and interface scheme of ISO 14649 based on the DIS-ballot version of September 2000 (International Standards Organization 2000).

2.1. *Information contents of ISO 14649*

ISO 14649, the input to the STEP-compliant CNC, is basically a structured feature-based representation of process plans for manufacturing processes such as milling operation, turning operation, rapid prototyping, etc. Currently, ISO 14649 is developed for milling operation based upon the geometric information of ISO 10303, such as AP 203, AP213, and AP 224. As illustrated in figure 1, the information contents of ISO 14649 are composed of: (1) *task description*, (2) *technology description*, (3) *tool description*, and (4) *geometry description*. *Task description* describes the logical sequence of executable tasks (e.g. *machining_workingstep*, *NC_function*) and data types. Details of each *workingstep* are covered in the *technology description* with reference to the *tool description* and the *geometry description*.

Specifically, the *workingsteps* include manufacturing features for 2.5D (*two5D_manufacturing_feature*) and 3D milling operations (*region*), and each *workingstep* has its subordinate sub-features (such as *planar_face*, *pocket*, *step*, *slot*, *round_hole*, and *general_outside_profile*) together with cutting condition information. It is important to note that the tool path specification in ISO 14649 is 'optional', unlike in the current NC programming where the tool path in terms of machine axes is the main information content.

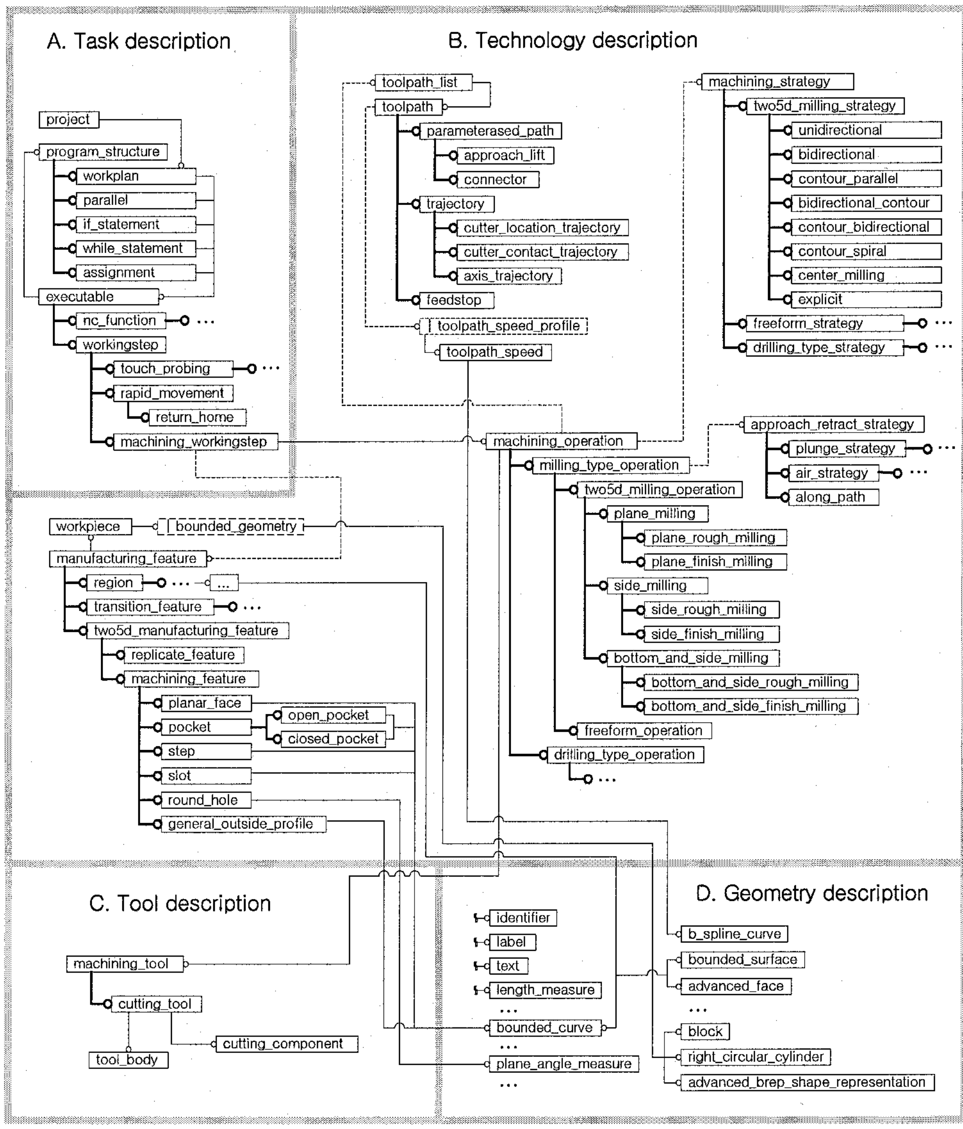


Figure 1. EXPRESS-G representation of ISO 14649.

2.2. *Interface of ISO 14649*

Compared with ISO 6983, ISO 14649 has a variety of information with a rather complex structure, whose programming is not an easy task if prepared manually. Thus, it is required to develop a computer-assisted part programming system. ISO 14649 code can be prepared by either the shop floor programming (SFP) system installed in the STEP-compliant CNC, or the off-line programming (OLP) system. In either case, a programming system should be developed so that appropriate information can be retrieved from the CAD/CAPP/CAM kernels. For instance, as illustrated in figure 2, the *geometry description* of ISO 14649 can be retrieved from the CAD kernel with a built-in AP 203 and AP 224 interface, *technology description* and *tool description* from the CAPP kernel with built-in AP 213 and ISO 13399 (International Standards Organization 1999) interface, and *tool path* from the CAM kernel. Based on the interface scheme with user interface modules, an ISO 14649 programming system, either in SFP or OLP fashion, can be developed.

In interfacing ISO 14649 code with CNC, the simplest way is via the post-processing module of the ISO 14649 programming system (OLP), converting ISO 14649 code into G-code. In this case, a conventional CNC can be used without modification. However, this cannot be considered as a STEP-compliant CNC since

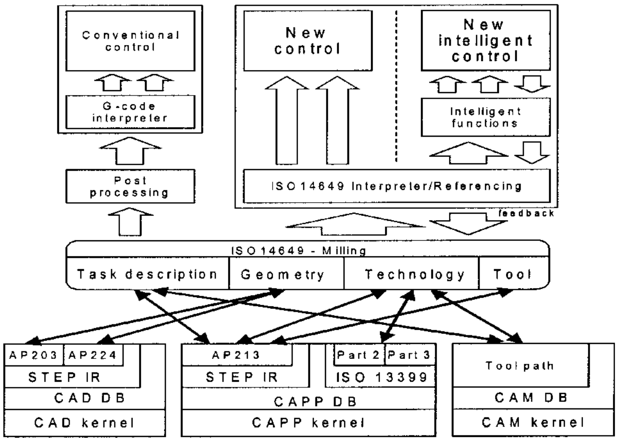


Figure 2. A programming and interface scenario of ISO 14649.

the STEP-compliant CNC should at least have the capability to read ISO 14649 code. Depending on how ISO 14649 code is utilized, the STEP-compliant CNC can be classified into two types. In the first, CNC ('New Control' in figure 2) faithfully follows ISO 14649 based on the sequence and tool path. The STEP-compliant CNC of this type does not have other intelligence than the capability to interpret ISO 14649 information contents.

The second type, which is more promising than the first is the 'New Intelligent Control' as shown in figure 2, where CNC is able to perform the machining task 'intelligently' and 'autonomously' based on the comprehensive information of ISO 14649. Considering that the purpose of ISO 14649 is not only to provide the information highway between STEP-based high level manufacturing functions (such as CAD, CAPP, CAM, PDM, ERP, MRP, etc), but also to provide rich information to CNC in order to make CNC intelligent, it is highly desirable to develop this second type, and this paper specifically deals with it.

3. Design considerations for STEP-compliant CNC

As described in section 2, ISO 14649 is a structured representation of fundamental information on how CNC carries out 'various intelligent functions'. Depending on the architecture, the effectiveness (practical gain) of ISO 14649 varies significantly. There are three critical considerations in designing the architecture of STEP-compliant CNC. First, ISO 14649 should be viewed as a 'guide map' (instead of as complete details) for the STEP-compliant CNC with room for flexibility to deal with unexpected changes occurring during the manufacturing process. In other words, the STEP-compliant CNC should be designed to have intelligent decision-making mechanisms using the information contents of ISO 14649 as references. Further, even if the tool path is specified (note that the specification of the tool path is optional in ISO 14649), the STEP-compliant CNC should be able to check the validity of the programmed contents at the time of execution with monitoring and recovery capability as well.

The second design consideration is concerned with the coverage of functions to be included in the STEP-compliant CNC. Since ISO 14649 is currently somewhat like 'uncooked raw materials with a rough recipe', it requires detailed recipes for different application domains. Currently, ISO 14649 has been developed mainly for milling process, on which this paper focuses, but other domains such as turning, EDM, grinding, RP, etc need to be included in the future. For intelligence and versatility, the STEP-compliant CNC should be designed to have:

(1) Comprehensive coverage of the tasks encountered during the entire milling process, such as part set-up, task planning, task execution, monitoring, and on-line inspection;
(2) Automated NC-code generation capability for the various tasks of part set-up, machining, on-line inspection, etc.;
(3) On-line adaptability to change occurring in manufacturing process; and
(4) Interface (communication) capability with external functions, such as off-line elements (CAD/CAM system, and shop-floor elements, such as CNCs, robots, etc) for cooperation.

Thirdly, an implementation strategy is also to be taken into consideration in designing the STEP-compliant CNC. To ensure the 'openness' of the STEP-compliant CNC, its functionality should be readily reconfigured by both the users and the developers. For such a purpose, the STEP-compliant CNC is required to take a software-based modular structure (e.g. written in C++ language) whose interface is made through a 'soft-bus' (e.g. via CORBA) instead of a hard-bus as in conventional CNC. This enables the system configuration to be built up in a 'Lego-block' concept so that functional modification can be readily made without involving a whole reconfiguration. Moreover, the soft-bus serves as a powerful platform for integrating the distributed modules implemented in different programming languages. Consequently, the compliant CNC has two conceptual buses: (1) ISO 14649 between programming system modules and STEP-compliant CNC, and (2) a soft-bus (e.g. CORBA and API) between the internal modules of STEP-compliant CNC, as shown in figure 3.

4. Architecture of an intelligent STEP-compliant CNC

Based on the discussion so far, an intelligent STEP-compliant CNC (called 'ASNC') is represented as shown in figure 4. Specifically, figure 4 shows a conceptual framework of ASNC addressing the implementation platform and functional modules required for autonomous operation, which can be reconfigured in plug-and-play fashion. ASNC is largely composed of four types of modules: (1) Basic modules: System Configuration, Graphic System, Database, (2) Functional modules: MMI (Man Machine Interface), Code Interpreter, Task Scheduler, Tool Path Generator, Operation Monitor, Setup Manager, On-line Inspector, (3) Interface modules: CORBA, Communication System, OS, etc., and (4) Control modules: NCK/PLC.

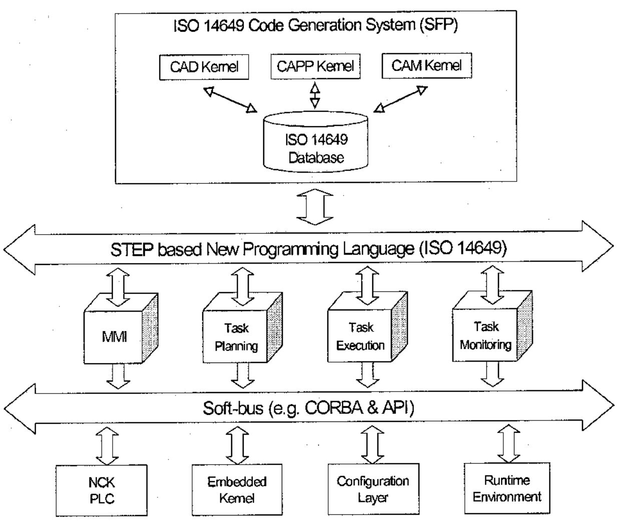


Figure 3. Two conceptual buses in STEP-compliant CNC.

To show how ASNC works, we present an operational scenario including functional details for major modules, as illustrated in figure 5. Note that the operational flow given in figure 5 is an example scenario emphasizing an autonomous operation in tool path generation and execution within CNC.

4.1. *Interpreter*

In the proposed ASNC, the first operation is to convert ISO 14649 information contents into the internal data format and process sequence in the form of a 'process sequence graph'. As illustrated in figure 6, the EXPRESS compiler interprets the physical file based on the schema for *task description*, *geometry*, *technology*, and *tool description*. The important function of the interpreter is to convert *task description* into the 'process sequence graph' (PSG) format.

PSG is a graphical representation of the sequence of *workingsteps* described in the *machining_feature* and *machining_operation* using the AND-OR relationship; i.e. each node represents *executables* connected by either AND or OR type of arc. AND is further divided into SA (Split-And) and JA (Joint-And), and OR into SO (Split-Or) and JO (Joint-Or). The OR relationship is specifically used to represent alternative process plans from which the ASNC (Scheduler) can choose to accommodate unexpected changes occurring on the shop floor.

It should be pointed out that the PSG scheme is our proposition for making the STEP-compliant CNC

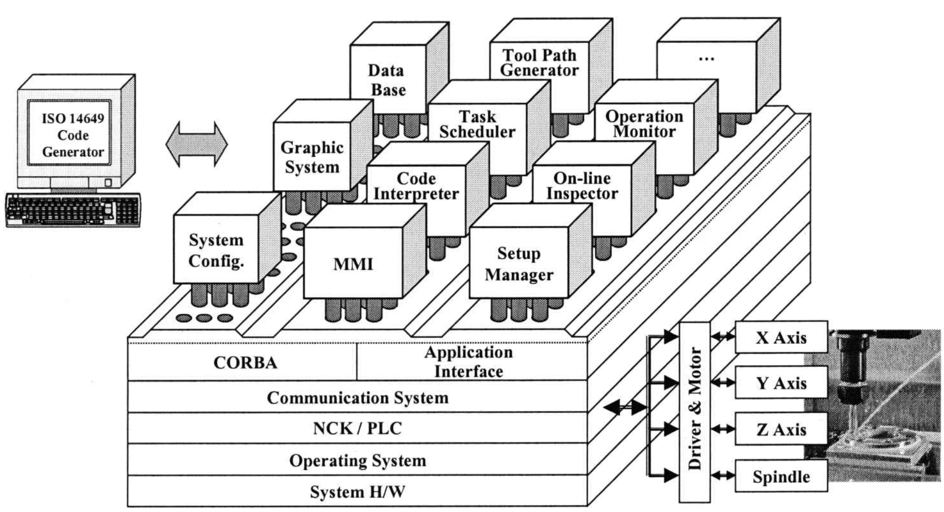


Figure 4. Conceptual architecture of an intelligent STEP-compliant CNC (ASNC).

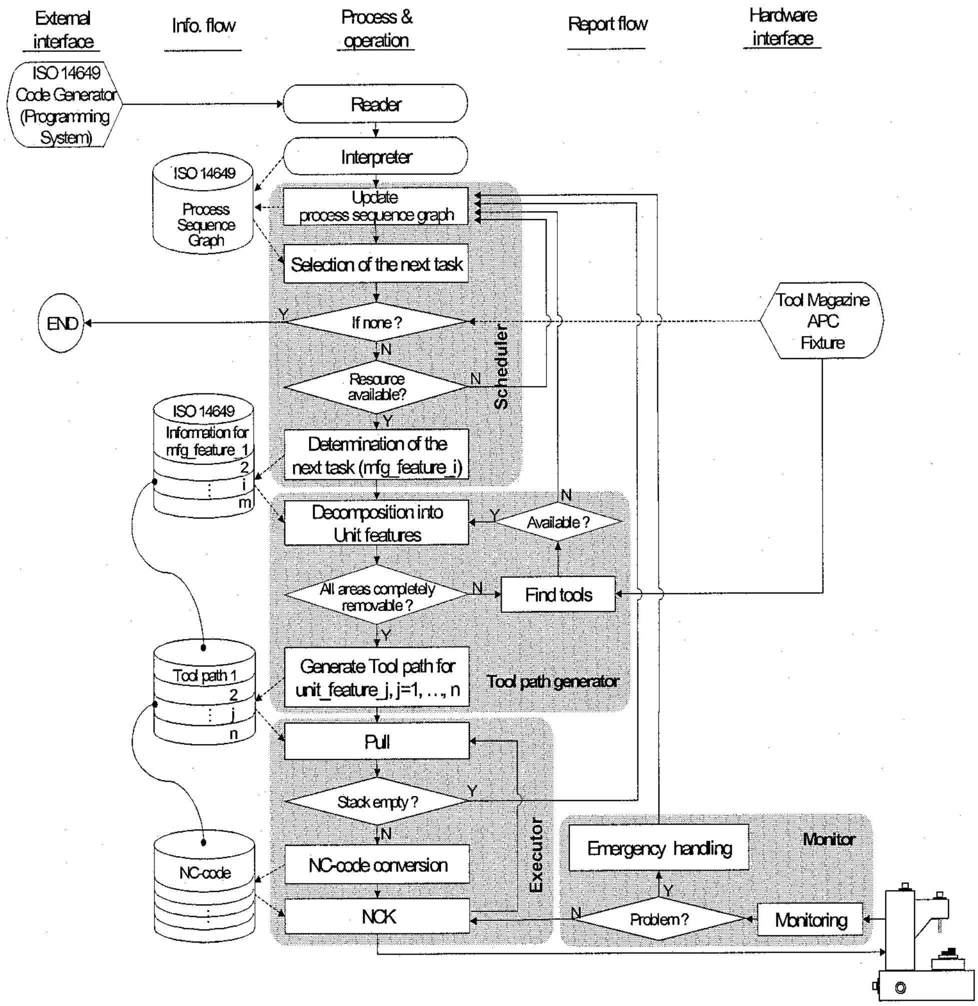


Figure 5. An operational flow of ASNC.

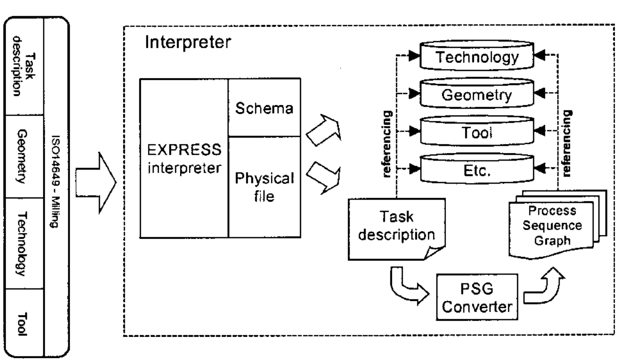
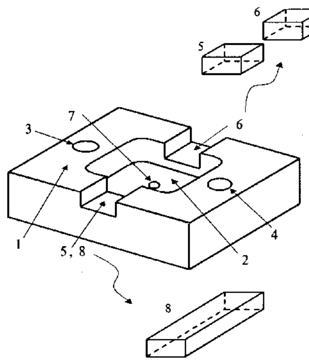


Figure 6. Interpreter function.

autonomous and intelligent. Since the current version of ISO 14649 allows only the description of a serial sequence of *workingstep*, it becomes inefficient as the number of alternative sequences increase, even if *if_statement*, *while_statement* schemas may be used for branching purposes. At the same time, to provide the STEP-compliant CNC with room for flexibility and optimization, ISO 14649 should also be able to accommodate a nonlinear process sequence scheme (instead of fixed sequence).

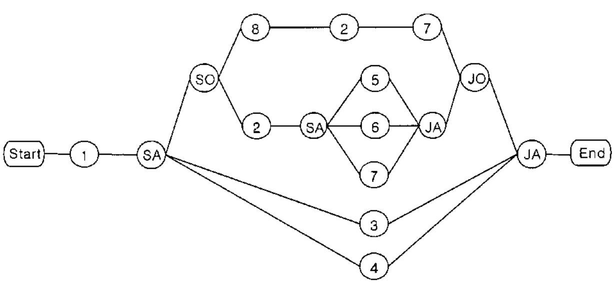
Consider an example part shown in figure 7(a). Suppose the manufacturing features and their machining operations are given by the programmer as shown in figure 7(b). Further suppose that the process sequence is represented by a new schema (e.g. Suh 2001), which is interpreted by the interpreter as shown in figure 7(c). This clearly shows the effectiveness of the PSG scheme. As indicated in figure 7(c), the part can be machined in various ways; e.g. 1-8-2-7-3-4, 1-3-4-2-6-5-7, 1-3-4-2-5-6-7, etc. By letting CNC choose the most appropriate method at the time of execution (for which



(a) An example part and its machining feature

WORKINGSTEP		
ID	FEATURE	OPERATION
1	planar_face	plane_rough_milling
2	closed_pocket	bottom_and_side_rough_milling
3	round_hole	drilling
4	round_hole	drilling
5	slot	bottom_and_side_rough_milling
6	slot	bottom_and_side_rough_milling
7	round_hole	drilling
8	slot	bottom_and_side_rough_milling

(b) *Workingstep* list



(c) Process sequence graph

Figure 7. Concept of nonlinear process plan.

the CNC should have selection criteria, such as the one given in section 4.2), the CNC can be endowed with flexibility, optimization, and adaptability, becoming far more intelligent and autonomous.

In addition to converting the new schema into PSG, the interpreter transforms manufacturing parameters for each *workingstep* (*mfg_data* in *technology description*, tool specs in *tool description*, and geometry information in *geometry description*) into an internal database, which is accessed by the Scheduler and Tool Path Generator to be described later.

4.2. *Scheduler*

The machining task given to ASNC is executed in terms of *workingstep* based on the sequence specified by PSG and the execution results reported by the Tool Path Generator and Monitor (see figure 5). The Scheduler functions: (1) to determine the next *workingstep* (node in PSG) to be executed, (2) to update PSG based on the report from the Tool Path Generator, Executor, and Monitor, and (3) to report the current status of machining task being executed upon requests from the external functions (such as CAD/CAM and other STEP-NCs for cooperative control).

The algorithm used to determine the next node is illustrated in figure 8. When a node is successfully completed (this is reported by Executor), the node is marked ‘C’ and the next node along the forward direction (toward END) is selected. When determining the next node, Scheduler checks the availability of resources (e.g. cutting tool in the tool magazine) required to machine the corresponding feature. If all the specified resources are available, the node is selected as the next task (see figure 5). Otherwise, the search process moves back until it hits SO (Split-Or) or SA (Spilt-And). If SO is found, the nodes between SO and JO are eliminated from PSG, and the next node is selected from the remaining SOs along the forward direction so that alternative process plan can be utilized. On the other hand, when SA is found, the Scheduler alarms the operator of the unavailability of resources, and marks the node ‘Q’ to indicate the node is in the queue until the required resources are available. In this case, the next node is selected among the unmarked ones.

The effectiveness of the scheduling procedure relies on how precisely and promptly the situation is reflected on PSG. PSG is updated based on the report from the Tool Path Generator, Executor, and Monitor in the following manner. When the current node under execution is successfully completed (this is reported by Executor), the scheduler marks the node ‘C’. However, when the current node is interrupted due to reasons such as tool breakage (reported by Monitor), it is marked ‘Q’, and the scheduler alarms to call for tool replacement. In case the machining volume corresponding to the node cannot be completely removed due to tool interference (this is reported by Tool Path Generator), the current node is decomposed into two: one part for machinable area, and the other for unmachinable area. Note that the Tool Path Generator checks the validity of the specified tool during the tool path generation, as described below. This happens when an improper tool is specified by the user.

4.3. *Tool path generator*

The Tool Path Generator (TPG) plays a crucial role in ASNC, and its primary function is to generate the tool path for the node specified by the Scheduler. Note that in ISO 14649, the specification of the tool path is optional. Even if the tool path is given, it may not be appropriate due to changes in machining environment (e.g. the specified tool may not be available in the tool magazine due to tool breakage). Thus, in ASNC the tool path is generated just before executing the assigned *workingstep*. The tool path is generated by utilizing most of ISO 14649 information; i.e. the *geometry description* of the manufacturing feature, *tool description*, and cutting conditions (*mfg_data*).

The tool path generation algorithms previously developed are hard to apply here, as they were based on a 'non-feature-based' approach. Thus, a new algorithm based on a feature-based approach needs to be developed. Since a detailed tool path algorithm is beyond the scope of this paper, the general methodology for feature-based tool path generation is outlined here (see Cho and Suh 1999 for a detailed algorithm).

Suppose the geometry of a manufacturing feature and the tool size are given. Then, the tool path can be generated by decomposing the manufacturing feature into 'unit' features, followed by determining the tool path for transition, approach, machining, and departure, as illustrated in figure 9. The unit feature is a fractional volume of the manufacturing feature defined by the radial depth of cut (RDC), axial depth of cut (ADC), and tool radius specified by *machining_operation*. Note that the unit feature can be removed by a single-path-tool-motion. Figure 10 provides some unit features and corresponding machining tool paths for the example given in figure 7.

If the entire volume of the manufacturing feature can be completely removed with the specified tool, the tool path is determined by connecting the tool path segments. Otherwise, a smaller tool is sought from the tool magazine, and the tool path is generated for the

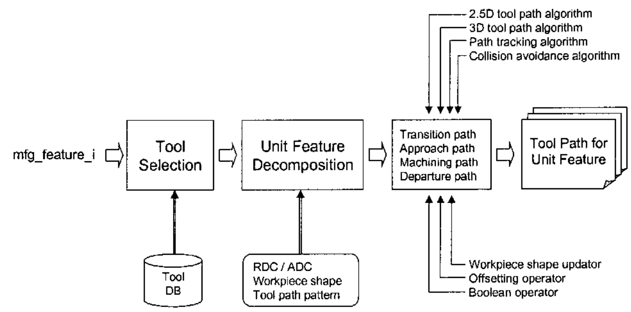


Figure 9. Tool path generation scheme.

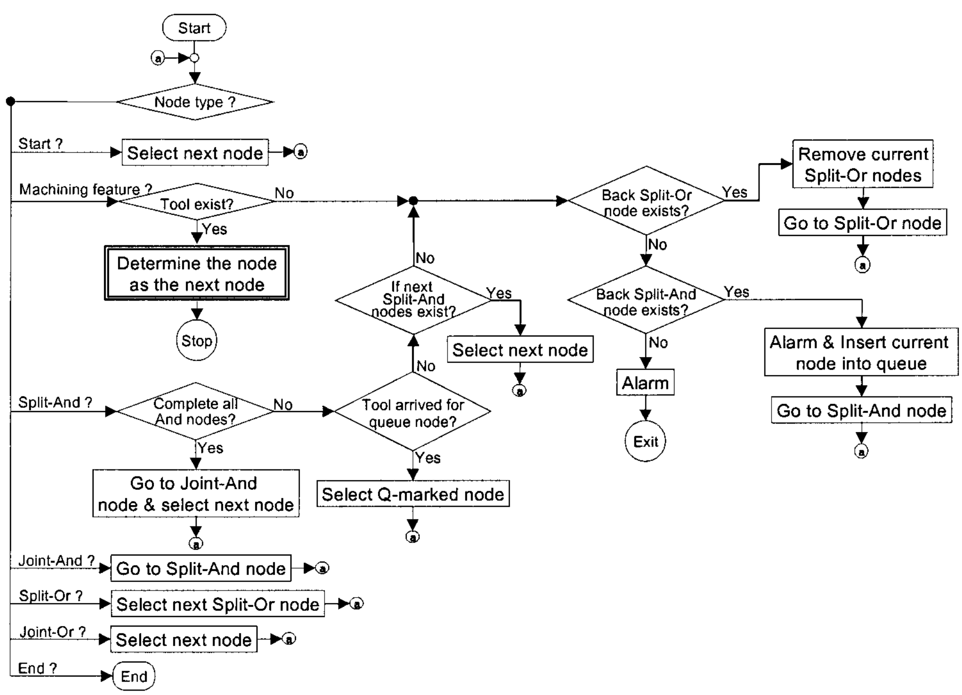


Figure 8. Determination of the next *workingstep* (node) in Scheduler.

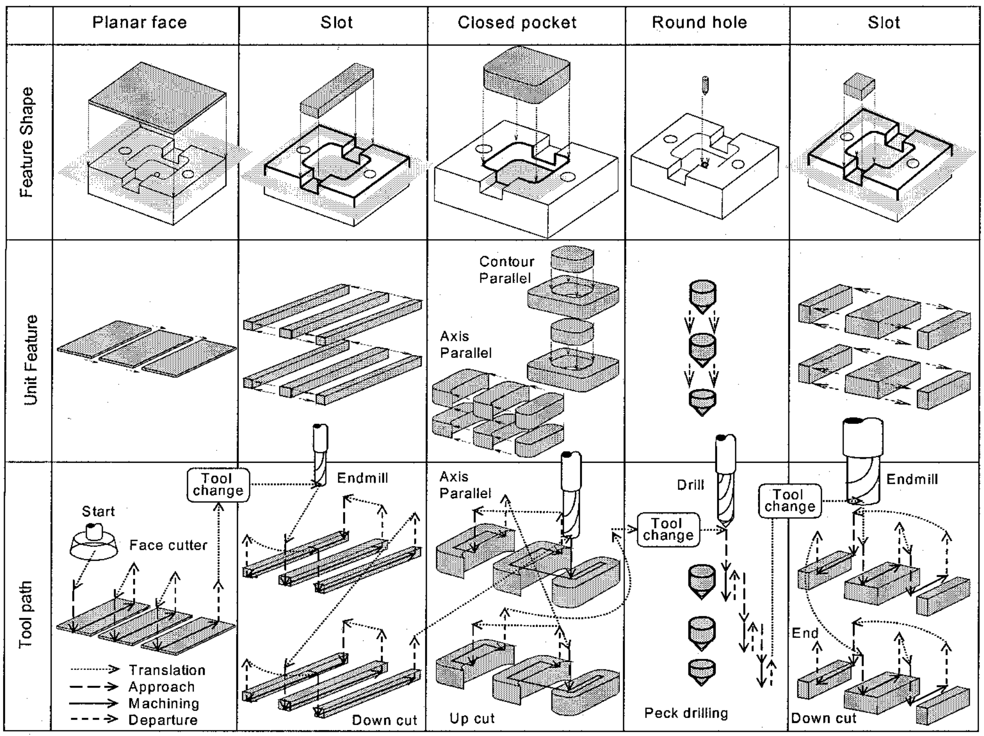


Figure 10. Unit features and their tool paths.

unremoved volume. In this way, ASNC ensures the validity of the tool specified in the input program, and checks the availability of the tool in the tool magazine before execution. The verified tool and tool path are stored in the tool path database for execution. Each time, the tool path segment for the unit feature is retrieved and converted to NC-codes for execution by NC kernel (NCK).

The TPG should be able to handle changes in process parameters, such as tool spec, RDC, ADC, etc for autonomous operation. Once the tool path is generated, ASNC executes the tool path in the order of unit features instead of line after line of NC instructions, as in conventional CNC. Thus, if the machining operation is stopped (due to tool breakage for instance) before completing the entire operation, ASNC is able to know where the motion is interrupted (tracing capability) in terms of the geometric feature instead of the machine coordinate. In such a case, ASNC generates a new tool path even if a different tool is required for removing the remaining volume.

4.4. *Executor and Monitor*

Noting the generated tool path stored in the tool path database in terms of unit features, Executor executes the tool path for the unit feature, first by pulling a unit tool path from the tool path database, followed by converting it into NC-codes. The converted NC-codes are stored in a NC-code database. The NC-code is pulled by NCK and executed by driving the hardware units (spindle, machine table, tool magazine, ATC, APC, coolant, etc). When the NC-code database is empty, the next tool path (of the unit feature) is pulled out from the tool path database until it is empty. When the tool path database is vacated, Executor reports to Scheduler that the current *workingstep* (node in PSG) is successfully completed (see figure 5).

The entire machining operation is under continuous surveillance by the Monitor. Depending on the aim of monitoring, various sensors are required together with fusion algorithms. For autonomous and automated operation, at the minimum tool breakage should be monitored. For optimal machining, an adaptive control scheme can be implemented. In the presented ASNC, only tool breakage detection is assumed, but more sophisticated monitoring functions can be incorporated. As indicated in the operational flow, when tool breakage is detected, an emergency action (e.g. retracting the tool) is taken immediately, followed by reporting to the Scheduler. By checking the tool path and NC-code databases, Scheduler knows the exact site of the problem in terms of unit features within *workingstep*. Based on this information, Scheduler takes appropriate action, as indicated in the scheduling algorithm of figure 8.

5. Concluding remarks

In this paper, we proposed an intelligent type of STEP-compliant CNC to enhance the effectiveness of CNC language (ISO 14649) by presenting a conceptual architecture and an operational scenario. The architecture is derived from the analysis of information contents of ISO 14649, and the role of CNC on the shop floor of an intelligent manufacturing system. It presents a picture for the next generation of CNC, which can be realized in the near future.

Based on the conceptual framework, we presented an architecture of intelligent STEP-compliant CNC called ASNC, together with an operational scenario and functional details. Throughout the operational scenario we showed that the new CNC can be 'autonomous' by using the new schema (PSG: process sequence graph) which addresses (1) the mission of each functional module, (2) the operation and information flow between the functional modules, and (3) the algorithmic issues for developing the functional modules. Furthermore, to achieve autonomy in a true sense, we suggested an update of ISO 14649 so that a nonlinear process sequence (PSG) can be also represented. We are currently developing ASNC based on the architecture presented in this paper.

Acknowledgements

This work was supported in part by a grant from the National Research Laboratory for STEP-NC Technology by MOST (Ministry of Science and Technology) in Korea, and grant (No. 1999-2-315-002-3) for interdisciplinary research program by KOSEF (Korea Science and Engineering Foundation).

References

Brouer, N. and Weck, M., 1997, Feature-oriented programming interface of an autonomous production cell. *Proceedings of the 4th IFAC workshop on Intelligent Manufacturing Systems*, Seoul, Korea, July, pp. 223–228.

Brussel, H., Valckenaers, P., Bongaerts, L. and Wyns, J., 1995, Architectural and system design issues in holonic manufacturing systems. *Proceedings of the 3rd IHAC workshop on Intelligent Manufacturing Systems*, Bucharest, Romania, November, pp. 1–6.

Cho, J. H. and Suh, S. H., 1999, On-line tool path generation and modification for STEP-NC. *Journal of CAD/CAM*, **4**(4), 295–311.

Chrysler, Ford and GM, 1994, Requirements of open, modular architecture controllers for applications in the automotive industry: version 1.1, http://www.arcweb.com/omac/Techdocs/omacv11.htm, December, 13.

ESPRIT, 1995, Open system architecture for controls within automation systems: Final report, ESPRIT III Project 6379, February.

Hardwick, M., 2000, Status report regarding the standardization of a new NC programming data interface, ISO TC/184/SC4 meeting, Charleston, SC, USA, 16 October.

HMSFPD, 1995, Holonic manufacturing systems: Full-scale project description. Holonic Manufacturing System Consortium, Sydney, August.

International Standards Organization, 1982, ISO 6983, Numerical control of machines program format and definition of address words.

International Standards Organization, 1999, ISO TC29, ISO 13399, Cutting tool data representation and exchange (DIS).

International Standards Organization, 2000, TC184/SC1/WG7, ISO14649, Data model for computerized numerical controllers (DIS-ballot version).

Kruth, J.P., 1994, A Prototype NC controller driven by feature-based part description. *Proceedings of the 4th Pacific Conference on Manufacturing*, Jakarta, Indonesia, December, pp. 535–544.

Liu, J. and Yamajaki, K., 1988, Agile production realization based on autonomously proficient CNC controller infrastructure. *Proceedings of the 31st CIRP International Seminar on Manufacturing Systems*, Berkeley, CA, USA, pp. 26–28 May, pp. 414–419.

OSEC Consortium, 1998, Development of OSEC (Open System Environment for Controller), OSEC-II project technical report, October.

Suh, S.H., Hong, H.D. and Cho, J.H., 1999, Feature based CNC control and message specification supporting HMS paradigm. CIRP Journal of Manufacturing Systems, 29(5), 437–441.

Suh, S. H., 2001, Propositions for future update of ISO 14649 based on DIS-ballot version. ISO TC184/SC4 Meeting, Funchal, Portugal, February 20.

Yamajaki, K., Hanaki, Y., Mori, M. and Tezuka, K., 1997, Autonomously proficient CNC controller for high-performance machine tools based on open architecture concept. *Annals of the CIRP*, **46**(1), 275–278.