Open access • Journal Article • DOI:10.1109/TASE.2005.862147

STEP-NC and function blocks for interoperable manufacturing — Source link

Xun Xu, Lihui Wang, Yiming Rong

Institutions: University of Auckland

Published on: 05 Jul 2006 - IEEE Transactions on Automation Science and Engineering (IEEE)

Topics: Computer-aided manufacturing, Electronic data interchange, Machine tool, Software portability and Interoperability

STEP-NC and Function Blocks for Interoperable Manufacturing

Xun W. Xu, Lihui Wang, and Yiming Rong

Abstract—**Interoperable manufacturing systems help manufacturing companies stay competitive in the environment of frequent and unpredictable market changes. An important part of a manufacturing system is computer numerically controlled (CNC) machine tools. Over the years, G-codes have been extensively used by CNC machine tools and are now considered as a bottleneck for making these machines adaptable and interoperable. Two new technologies emerged in recent years: Standard for the Exchange of Product data for Numerical Control (STEP-NC) and function blocks. The STEP-NC data model represents a common standard for NC programming, making the goal of a generic NC code generation facility a reality. Function blocks are an emerging IEC standard for distributed industrial processes and control systems. They can be used for CNC controls to encapsulate machining data, such as machining features and their needed algorithms. This paper introduces the above two new standards and the technologies that are developed based on the standards. The main body is devoted to analyze the standards from the functionality viewpoint. These functionalities include, bidirectional information flow in computer-aided design/computer-aided manufacturing, data sharing over the Internet, the use of feature-based machining concept, modularity and reusability, intelligent and autonomous CNC, and portability among resources. Some implementations are also presented to showcase how the standards are used to develop technologies for interoperable machining.**

Note to Practitioners—**Modern computer numerically controlled (CNC) machine tools are limited in functions because their controllers rely on G-codes for communications. G-code is considered a "dumb" language as it only documents instructional and procedural data, leaving most of the design information behind. G-code programs are also hardware dependent, denying modern CNC machine tools desired interoperability and portability. In recent years, two new standards emerged, STEP-NC and function blocks. They may hold the key to empowering CNC machine tools with richer information which, in turn, gives CNC machine tools the ability to "think" intelligently and to be interoperable. This paper introduces these two standards, the technologies that have been developed based on the standards and some prototype systems using the standards and technologies. The intention is not to highlight any achieved research outcome. Instead, the focus is on informing the research and practical world about these new standards, analyzing them from the viewpoint of supporting interoperable CNC machine tools, and offering some futuristic views about these standards and technologies. While these standards are still in their infancy, research activities and prototype systems are already coming thick and fast. There seems to be a "healthy" mixture of participants working in the field. They range from the manufacturers of all systems related to the data interface (i.e., CAM systems, controls, and machine tools), to the users and academic institutions.**

Index Terms—**Computer numerical control (CNC), function blocks, interoperability, Standard for the Exchange of Product data for Numerical Control (STEP-NC).**

I. Introduction

THE LAST-HALF century has seen a number of revolutionary changes to a manufacturing system's configurations. The traditional configuration of a manufacturing system was the dedicated transfer (machine) line. It enables mass production at high efficiency and low cost. During the 1980s, "flexible" manufacturing was developed to meet the need for the production of smaller batches of different parts. These systems used groups of computer numerically controlled (CNC) machines that could be reprogrammed to make different parts. These machine tools have since become the central elements in systems, such as flexible transfer lines, flexible manufacturing cells (FMC), and flexible manufacturing systems (FMS).

However, the flexibility existing in these systems, in particular, the flexibility of CNC machine tools, is still believed to be limited. The current CNC control regime is considered as rigid and proprietary. This is largely due to the continued use of the out-of-date CNC programming language—ISO 6983 [1] (RS 274D), or commonly known as G-codes. This language focuses on programming the path of the cutter center location (CL) with respect to the machine axes, rather than the machining tasks with respect to a part. Thus, ISO 6983 defines the syntax of program statements but, in most cases, leaves the semantics ambiguous. These programs, when processed in a computer-aided-manufacturing (CAM) system by a machine-specific postprocessor, become machine dependent. There is limited control over program execution. This type of CNC control regime means that the output from a CAM system has no adaptability which, in turn, denies the CNC machine tools from having any interoperability.

In order to prepare manufacturing companies to face increasingly frequent and unpredictable customer demands with confidence, interoperable CNC machining systems may hold the key to the solutions. This type of system can be characterized as being capable of 1) seamless information flow; 2) feature-based machining; 3) autonomous CNC; 4) fault tolerable; 5) networked and distributive; 6) modular; 7) scalable; 8) portable; and 9) adaptive. Anything but a lack of such a system means that the manufacturing industry is fighting a losing battle in keeping up with the new market trends. The automotive interoperability study estimates that the U.S. automotive industry

Manuscript received March 24, 2005; revised August 11, 2005. This paper was recommended for publication by Associate Editor S. Gupta and Editor P. Ferreira upon evaluation of the reviewers' comments.

X. W. Xu is with the Department of Mechanical Engineering, University of Auckland, 92019 Auckland, New Zealand (e-mail: x.xu@auckland.ac.nz).

L. Wang is with the Integrated Manufacturing Technologies Institute, National Research Council Canada, London, ON N6G 4X8, Canada (e-mail: lihui.wang@nrc.gc.ca).

Y. Rong is with the Department of Mechanical Engineering, Worcester Polytechnic Institute, Worcester, MA 01609-2280 USA (e-mail: rong@wpi.edu).

Digital Object Identifier 10.1109/TASE.2005.862147

has to spend about U.S.$10 billion a year to overcome information and system barriers and poor interoperability among computer-aided-design (CAD), computer-aided process planning (CAPP), and CAM systems [2]. About U.S.$9 billion is used to repair and replace unusable data files.

Research into manufacturing interoperability has been underway for some years on issues at the system level as well as the component level (i.e., machine and control). At the system level, the development work includes the tools and methodologies to design a manufacturing system and evaluates various configurations [3]–[8]. Among these methodologies, agent-based tools and techniques have proven to be a popular and effective approach [5]–[7]. At the component level, research work has primarily centered around the control issues. The aim is to increase today's CNC system's flexibility, particularly the trend toward open architecture control, based on MDSI CNC architecture [9], open system architecture for controls association (OSACA) [10] Open System Environment for Controller (OSEC) [11] and open modular architecture controller (OMAC) [12] where third-party software can be used at the controller working within a standard operating system. One further recognizable industrial development is the application of software controllers, where PLC logic is captured in software rather than in hardware. These developments have improved software tools and the architecture of CNC systems to a limited degree and in limited aspects. Vendors and users are still seeking language and data modeling tools that unify CAM/CNC with the upstream applications, such as CAD/CAPP, and more important, enablling interoperable machining.

In a broader scope, research work has been carried out to facilitate information flow across the CAD, CAPP, and CAM domains. The main objective is to provide an integrated, interoperable, and extendable platform for design and manufacturing activities. The National Institute of Standards and Technology (NIST) has developed an open, neutral manufacturing process object model to enable software interoperability among preliminary product design, process planning, and manufacturing execution [13]. This object model has been used as the basis for developing the ISO 16 100 standard: Industrial automation systems and integration—Manufacturing software capability profiling, Part 2: Information models for interoperability. Zhao, *et al.* reported a cooperative agent model for process planning [14]. The model made use of intelligent agents and tried to satisfy five major requirements simultaneously: autonomy, flexibility, interoperability, modularity, and scalability. An EXPRESS model for sharing and exchanging flexible manufacturing resources data has been developed [15]. The aim is to overcome the incompatibility and interoperability problems among the proprietary formats found in CAD, CAPP, and CAM. The commonality among these systems and those of the similar nature is that they fall short in addressing the CNC programming language and its control regime.

At the turn of the 21st century, two new standards emerged, ISO14649 recognized informally as STEP-compliant numerical control (STEP-NC) [16]–[20] and function blocks [21]. These two standards and a new breed of CNC systems that these standards promise to support may hold the key to achieving interoperable CNCs. The STEP-NC is being developed under the ISO Technical Committee TC184 in the subcommittees SC1 and SC4 in combination with the support of the STEP-NC IMS project in Europe and Asia, and the super model project in the U.S. Contrary to the current NC programming standard (ISO 6983), STEP-NC is not a method for programming. Nor is it intended for describing the tool movements (e.g., axis motions) for a specific CNC machine. Instead, it provides an object-oriented data model for CNCs with a detailed and structured data interface that incorporates feature-based programming. Function blocks are an emerging International Electrotechnical Commission (IEC) standard for distributed industrial processes and control systems. It is based on an explicit event-driven model and provides for data flow and finite-state automata-based control. Based on previous research on function blocks [22], function blocks can be used as the enabler to encapsulate process plans, integrate with a third-party dynamic scheduling system, monitor the process plan during execution, and control machining jobs under normal and abnormal conditions. They are suitable for machine level monitoring, shop-floor execution control and CNC control. The combination of the two can be seen as a "natural marriage." This is because the former provides an informationally complete data model but with no functionality, whereas the latter can embed intelligence and provide functionality in the data model for a more capable CNC regime.

This paper introduces the above two new technologies with respect to interoperable CAPP, CAM, and CNC in particular. While potentials are being recognized for both technologies to become enabling tools for interoperable manufacturing, there are strengths and weaknesses between them to a varying extent. It is of importance to understand these differences before embarking on any implementations. The main body of this paper is devoted to analyze and contrast the technologies from the functionality viewpoint. The functionalities that are discussed herein include bidirectional information flow in CAD/CAM, data sharing over the Internet, use of the feature-based machining concept, modularity and reusability, intelligent and autonomous CNC, ease of integration with other modules, and portability among resources (machines). All of them are essential characteristics of an interoperable CNC system. Some prototype systems based on STEP-NC and function blocks are also presented. This is to showcase how these technologies may be used for interoperable machining.

II. STEP-NC—STEP-COMPLIANT NUMERICAL CONTROL

As STEP-NC is an extension of STEP to handle NC processes, it strictly follows the STEP standard. This means that all of the implementation methods [23]–[31] defined in STEP for building and exchanging product models apply. Part 21 physical file is the most common type of interface mechanism. In a STEP-NC file, for example, the HEADER contains general information and comments about the part program, such as the filename, author, date, and organization. The DATA section is the main section of the program, containing all of the information about manufacturing tasks and geometries. The data are divided in three significant parts: 1) workplan and executables; 2) technology description; and 3) geometry description. This section must also include a PROJECT entity that is an explicit reference for the starting point of the manufacturing tasks. The PROJECT entity contains a main workplan that, in turn, contains sequenced executable manufacturing tasks or commands. The order of execution of manufacturing operations is given by the order of executables. In order to change the sequence of operations, only this part of the program file has to be changed. The workplan combines several executables in a linear order or depending on given conditions if conditional controls are used.

In essence, STEP-NC describes "what-to-make," whereas the G-code describes "how-to-make." STEP-NC describes tasks (e.g., predrilling, drilling, roughing, and finishing) that are based on the machining features (Fig. 1) so that the part program supplies the shopfloor with higher-level information (i.e., the information about machining tasks and technological data on top of pure geometrical and topological information). As a result, modifications at the shopfloor can be saved and transferred back to the planning department which enables a better exchange and preservation of experience and knowledge.

There have been a number of benefits recognized [33]. STEP-NC provides a complete and structured data model, linked with geometrical and technological information, so that no information is lost between the different stages of the product development process. The data model is extendable to further technologies and scalable (with conformance classes) to match the abilities of a specific CAM, shopfloor programming (SFP), or NC. The postprocessor mechanism will be eliminated as the interface does not require machine-specific information. Modification at the shopfloor can be saved and fed back to the design department, hence, bidirectional information flow from CAD/CAM to CNC machines can be achieved.

Currently, two versions of STEP-NC are being developed. The first is the application reference model (ARM) [16]–[19] and the other application interpreted model (AIM) [20]. The ARM, ISO 14 649 entails a detailed analysis of the requirements of CNC applications. The specific objects are accurately defined as are the relationships among them. The structure is written in a data modeling language called EXPRESS, which is defined in the standards as a description method [17]–[19]. On the other hand, the AIM model of STEP-NC (i.e., STEP AP-238), is a way to map the application requirement data stipulated in the ARM (ISO 14 649), using a fixed set of STEP concepts called "generic resources," into an integrated ISO 10 303 application protocol. Because AP-238 is built on the same foundation as the other STEP application protocols, it can share data seamlessly. For more information on the use and differences between them, readers are referred to [34] and [35].

The global research in the area of STEP-NC has remained highly visible with a number of major projects coordinated and conducted across different countries in the same region as well as on a truly international scale [36]. There are three types of projects, those carried out a) on the international scale, b) across a few countries in the same region, and c) within a country. On the international scale, the IMS STEP-NC project [37], [38], endorsed in November 2001, entails a true international package of actions with research partners from four different regions: European Union, Korea, Switzerland, and the U.S. They covered manufacturers of all systems

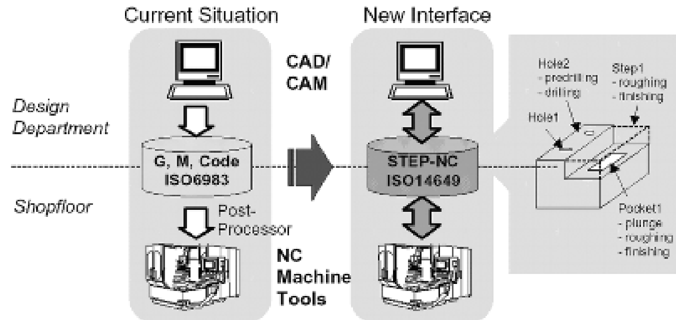

Fig. 1. Comparison of ISO 6983 and ISO 14 649 [32].

related to the data interface (i.e., CAM systems, controls, and machine tools), the users and academic institutions. The regional coordinators are Siemens (EU), CADCAMMation (Switzerland), STEP Tools (U.S.), and ERC-ACI (Korea). Siemens is also the inter-regional coordinator. Because of the Workingstep-based feature in STEP-NC, many projects have incorporated prototype/commercial CAPP or CAM systems with process planning functionalities (i.e., STEPturn from ISW Stuttgart, Germany [39]; the WZL-SFP from WZL Aachen, Germany [38]; ST-Plan from STEP Tools Inc., U.S. [40], [41]; the SFP system from the National Research Laboratory for STEP-NC Technology (NRL-SNT), Korea [42], [43], the AB-CAM system from the Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, U.K. [44], [45], and *STEPcNC Converter* from the University of Auckland, Auckland, New Zealand [46]).

III. Function Blocks

The IEC-61 499 function block specification [21] is an emerging IEC standard for distributed industrial process measurement and control systems (IPMCS). It is based on an explicit event-driven model and also provides data flow and finite-state automata-based control. It is thus relevant to distributed CNC machining for interoperable manufacturing. Being an atomic distributable and executable control function unit, a function block instance can encapsulate a part of machining process data (e.g., slot roughing, pocket finishing, and hole drilling, etc.) for a given machining feature [47]. It is comprised of an individual, named copy of the data structure specified by its function block type, which persists from one invocation of the function block to the next. Fig. 2 illustrates both the basic (left) and composite (right) function blocks.

A function block, especially the basic function block, can have multiple outputs with internal state information hidden. This means a function block can generate different outputs even if the same inputs are applied. These characteristics are of vital importance for automatic cutting condition generation/modification after an NC program has been downloaded to a CNC controller, by changing the internal state of a function block (a machining process). For example, an NC program for the same pocket roughing can be used by two different milling machines with different cutting conditions, simply by adjusting the internal state variable of the function block instances. Similar to object-oriented definitions, a function block type can be considered as a class, and a particular function block is the instance

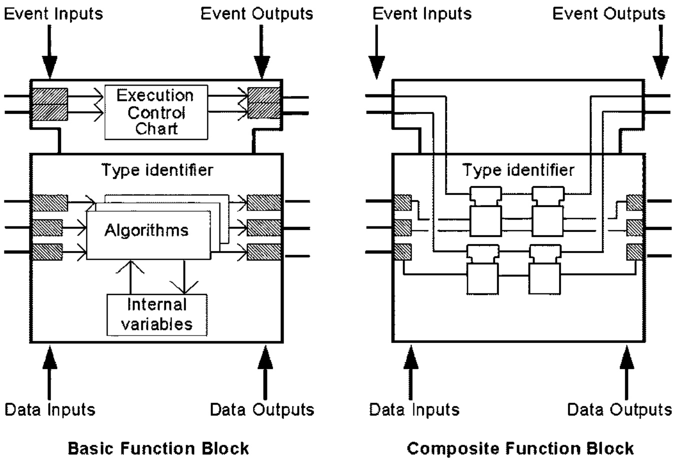

Fig. 2. Function block structure.

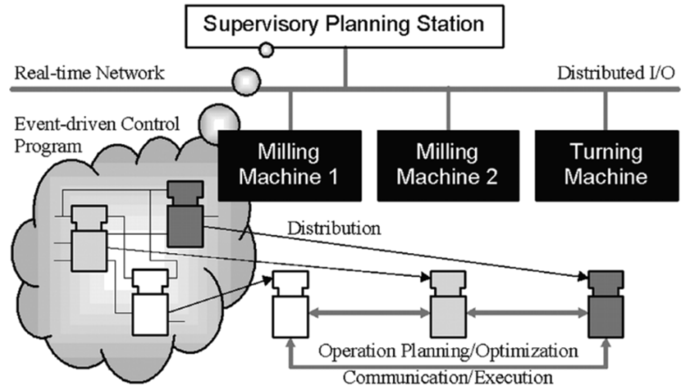

Fig. 3. Function block distribution [48].

of that class. For example, a pocket milling function block can be used for either roughing or finishing depending on the message received. However, different from the object-oriented approach, the behavior of a function block is controlled internally by a state machine whose operation can be represented by an execution control chart (ECC). Each basic function block is an atomic unit for execution. Composite function blocks may require a multi-threaded concurrent execution model due to the complex event flow. The event flow determines the scheduling and execution of machining operations specified by the algorithms (methods) in basic function blocks. It can also provide signals to maintain or change the internal state variables. In terms of machining process encapsulation, basic function blocks encapsulate both the data and functions of machining process, whereas composite function blocks only encapsulate basic function blocks.

Not only can function blocks encapsulate machining processes, they can also provide support for their communications. The ECC for the state machine of a function block can be used to control their internal algorithms [47]. It is anticipated that in the future, CNC controllers may have function blocks as part of their device firmware or provide function block libraries from which function blocks can be selected and downloaded. However, to better utilize the legacy machine tools, vendor or user extensions to the standard G-codes could be encapsulated by function blocks and distributed to end users to be directly utilized by their machining applications.

Furthermore, function block-based process plan encapsulations enable and facilitate transparent distribution and dynamic scheduling of machining processes over a group of machine tools. As shown in Fig. 3, a combined process plan represented by three function blocks may be assigned to a single machine controller or may be distributed over several machine tools, depending on the process plan and the availability of each machine, with no interruption to the entire machining operation. The communication (event or data flow) between the function blocks can be realized by a well-established message passing mechanism, such as eXtensible Markup Language (XML), to keep machining tasks on schedule. This capability is crucial for improving system performance and for rapid fault recovery in a machining shopfloor, where run-time process planning and dynamic rescheduling can be done in parallel whenever a run-time exception occurs. In addition to basic and composite function block types, the service-interface (SI) function block is to provide an interface between function blocks and resources (machines) for data/event communication. In fact, wherever any form of interaction is required between function blocks within the resource and the external world, there is a requirement for an SI function block. Such interactions include reading values of current cutting parameters, setting a spindle speed, publishing machining status for monitoring, and facilitating dynamic scheduling.

IV. STEP-NC versus Function Blocks as Technology Enablers for Interoperable Machining

An in-depth look at STEP-NC and function blocks in terms of their supporting interoperable manufacturing is presented in the following sections. An effort was made to present to the readers with an unbiased view of the technologies in the following eight areas of interest, in terms of how each one of them can: 1) support bidirectional information flow in the design and manufacturing chain; 2) adopt the concept of feature-based machining for CNCs so that higher-level information can be made available at the CNC machines; 3) enable an autonomous and intelligent CNC, most likely within the CNC controller and at run-time; 4) provide a fault-tolerable feature for CNCs; 5) enable data sharing over the Internet and support a distributed process planning scenario; 6) offer features such as modularity, reusability, and openness to the CNC systems; 7) be scalable, tailorable, and extensible; and 8) empower CNCs with portability among various manufacturing resources.

A. *Supporting Bidirectional Information Flow in CAD/CAPP/CAM/CNC*

Current solutions of data exchange, such as Standard d'Echange et de Transfert (SET), Verbandes der Deutschen Automobilindustrie (VDA), and International Graphic Exchange Specification (IGES) [49] are partially successful [50] and not totally suitable to all of the needs of the manufacturing industry. STEP-NC defines a new interface for an effective, bidirectional information exchange between CAD/CAM systems and NC controllers. Its feature-based structure is fully compatible with the widely disseminated STEP standard.

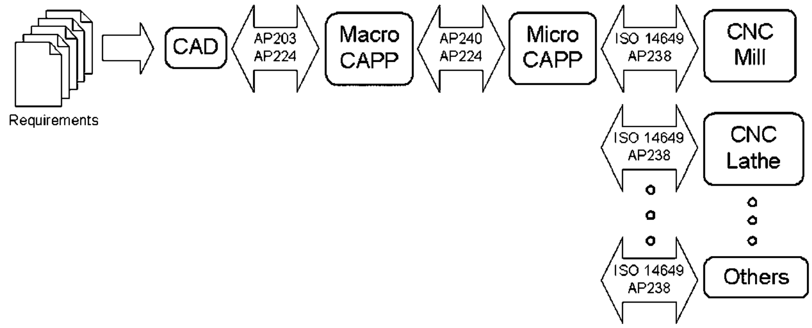

Fig. 4. Bidirectional information flow in CAD/CAPP/CAM/CNC [36].

Modification at the shopfloor can be saved and fed back to the design department, hence bidirectional information flow between CAD/CAPP/CAM and CNC can be achieved (Fig. 4).

As the AIM of STEP-NC, AP-238 is built on the same foundation as the other STEP application protocols so data sharing with those protocols is made even more straightforward. Take an example where AP-203/AP224 data are built by a CAD system, sent to a CAM system for conversion into AP-238, and sent to a CNC for manufacturing. The CAM system does not need to redefine the geometry made by the CAD system because the two APs share the same data. The CAM system only adds process information to the CAD data so that the CNC has enough input to make the part. When modifications are made, for instance, to a machining operation which, in turn, alter the design model, all of the changes can be easily preserved and fed back to the design system.

With function blocks being initially designed as a new PLC language, they are more applicable to the device-level communication and logic control. In the context of interoperable manufacturing, function blocks have limited support for bidirectional information flow in the CAD/CAPP/CAM/CNC chain. Their bidirectional information flows exist mainly between shopfloors and CNC controllers for process monitoring, dynamic scheduling, and execution control.

B. Use of Feature-Based Machining Concept

The significance of the "feature" concept, or rather its implication and implementation in design and manufacturing domain, has long been recognized. Most of the contemporary CAD/CAM systems have all adopted the feature-based design approach. On the CNC side, feature-based machining still remains as a much-anticipated reality. Central to the concept of feature-based machining is effectively that of "task-oriented" as opposed to the "method-oriented" CNC programming. G-code documents CNC machining instructions, which are rigid and machine dependent. STEP-NC, on the other hand, brings task-related information to a CNC machine. In STEP-NC, the feature-based machining concept is embedded in a well-established hierarchy of workingstep supertypes and subtypes. It breaks down every machining operation into steps required to perform the operation. These steps include actions to be taken for each of the machining features defined. In other words, the "workingstep"-based structure replaces the axis motion and tool operation command sets as in G-code. STEP-NC

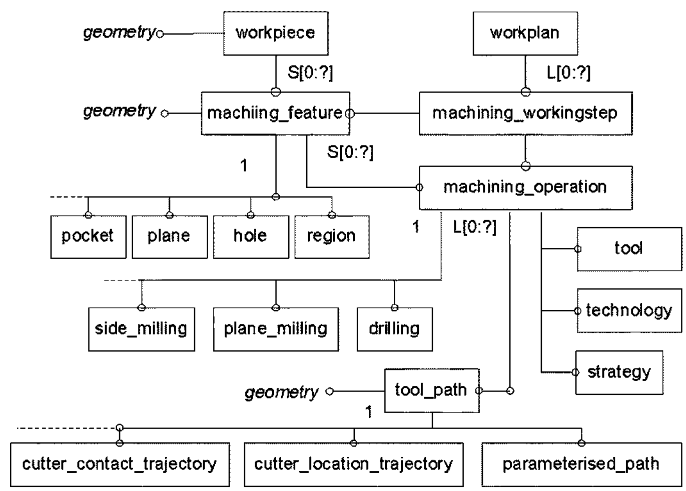

Fig. 5. Feature-based STEP-NC data structure [32].

provides a feature-based and object-oriented data model for CNCs. It is systematically detailed with a structured data model (Fig. 5) that incorporates the component geometry through feature-based programming.

The Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University has been working on the 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 [51]–[54]. A graphical user interface has been developed, using Shop Mill (a shopfloor-oriented NC programming tool) and Sinumerik 840D HMI (Human Machine Interface). This interface can parse feature-based STEP-NC data [52]. The same team has also developed its own STEP-enabled CNC controller called WZL-NC. WZL-NC can control a five-axis Maho600E machine tool [38].

Similar to STEP-NC, function blocks may also be coupled with machining features. Each function block type corresponds to one machining feature class. By selecting an appropriate function block, a process planner implicitly tells a CNC controller "what to do" and leaves the "how to do" to the algorithms embedded in the function block. In other words, a function block knows how to fabricate a defined machining feature with which it is coupled. As shown in Fig. 6, the concept of function blocks can be easily integrated with STEP-NC through Workingsteps. From this perspective, function blocks can be used as a new class of NC control language to replace the G-codes.

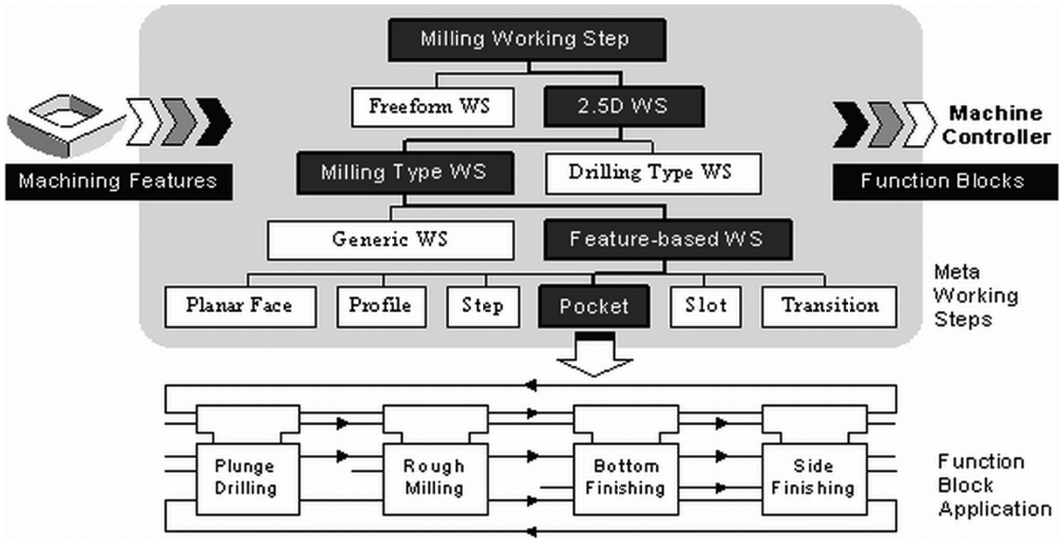

Fig. 6. From machining features to function blocks [55].

C. Autonomous and Intelligent CNC

The autonomous functions with a CNC machine are those that allow the reconfiguration of machining tasks at the workstation with little human intervention. An intelligent CNC controller is expected to self-prevent errors, optimize operations, and dynamically recreate tool paths. As ironic as it may be, the modern CNC machines, on one hand, can perform extremely complex operations at a high production rate, yet they have little "thinking" ability when executing machining commands. This is, of course, due to the use of the low-level CNC language—G-code. Due to this, the current practice has been that machining optimization and regeneration of the part programs based on resource availability are dealt with by CAPP systems. Various methodologies have been suggested for optimizing process plans taking into account complex constraints present in the domain [56], [57].

STEP-NC supports autonomous and intelligent CNC by providing complete information about machining tasks to the CNC controller. Machine tools can have "bigger brains" to intelligently plan and replan their operations using a rich set of data about the product and process both offline and online. With such information made available with a STEP-NC controller, a number of high-level tasks can be performed. A part program can be optimized and regenerated based on the information such as tooling, which is available at the time of manufacturing rather than at the time of process planning. Run-time decision making is possible to accommodate machining parameter variations, such as tool length and radius compensation, compensation of time-dependent tool wear and thermal deformations, real-time machining parameter adjustment to combat chattering, and the dynamic recreation of a tool path. Some research work has been carried out toward the realization of autonomous and intelligent CNC. An autonomous STEP-compliant CNC (ASNC) [58], [59] has been developed at the National Research Laboratory for STEP-NC Technology, Pohang University of Science and Technology, South Korea.

Different from G-codes, function blocks enable autonomous machining processes by embedding intelligence in them in the form of "algorithms." They are event (or resource) driven. For a given machining feature and a set of specified resources (machine and cutting tool, etc.), these algorithms are able to decide at run time how to machine the feature using the available resources. The run-time decision making includes cutting parameter optimization, tool-path generation, and motion control. For the sake of legacy machine tools that only recognize G-codes, a function block can output G-code after local optimization. It needs to be pointed out that more often than not, most of the complex optimization tasks are carried out through composite function blocks instead of basic function blocks [60].

Fig. 7 depicts the function block design and embedded machining information of a typical machining feature—a four-side pocket [61]. The algorithms ALG_INI, ALG_ROU, ALG_FIN, and ALG_MON are responsible for initialization, roughing, finishing, and machining process monitoring, respectively.

D. Fault Tolerable

The fault-tolerable feature on a CNC machine is the capability of a higher-level robustness and self-correction capability exhibited by the controller when unexpected events (e.g., abnormal tool breakage and tool wear, and the presence of workpiece defects) occur during execution of machining commands. STEP-NC has no specific functions to enable this feature on a controller. However, for a controller to be fault tolerant, it needs to be "mindful" throughout the entire duration of program execution as to which feature is being machined, where it is in machining the feature, the geometric information of all the machining facilities involved, and what possible actions there are to take at any stoppage. A STEP-NC data model captures and provides this information for a controller. The autonomous and intelligent features that the STEP-NC enables (e.g., run-time regeneration of toolpath) as described in the previous section, provide the necessary leverage.

As mentioned in the previous section, function blocks are capable of making decisions at run time and they are resource driven, meaning that the function blocks can adapt themselves to environment changes, such as alternative tools or machines due

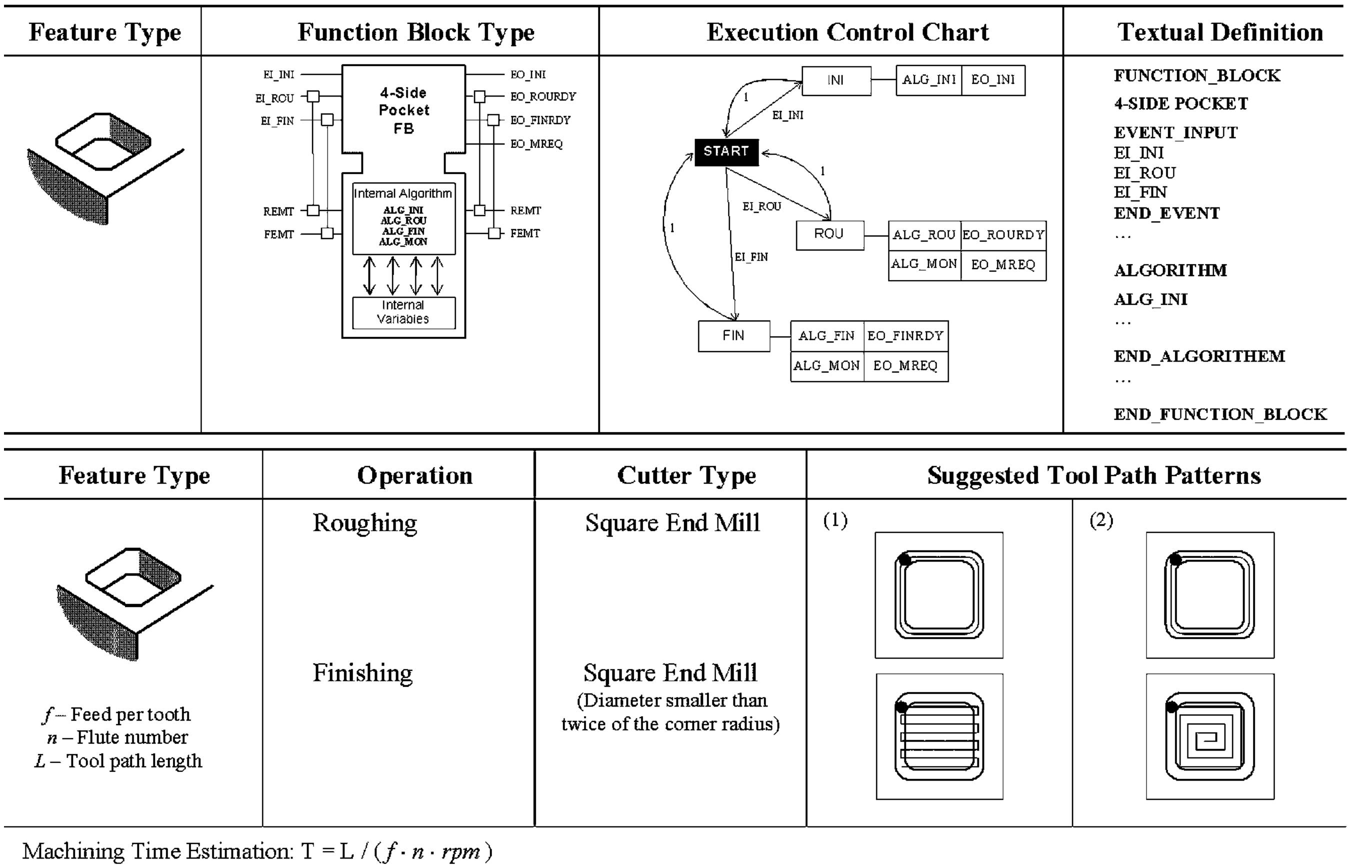

Fig. 7. Embedded machining information of a function block [61].

to tool breakage or machine breakdown. Not only can a function block optimize the cutting parameters and tool path for an alternative cutting tool, it can also be easily redirected to another machine for the rest of machining for fault recovery. Because of the self-monitoring feature, function blocks can resume machining operations right from where they were interrupted.

E. Data Sharing Over the Internet—Capability of Working in a Distributed Network

The ability of sharing and exchanging data over the Internet with few barriers is a prerequisite for a suite of newly emerged manufacturing technologies, for example, distributed process planning, collaborative product development and manufacturing, virtual machining and even e-Manufacturing. STEP-NC can support a distributed and collaborative manufacturing scenario by means of Extensible Markup Language (XML), or rather XML binding, known as Part 28 [28], [29]. The first methodology suggested is to map EXPRESS defined data into XML data defined by document-type definition (DTD) as described in XML 1.0. Owing to the soon recognized limitation, a new language has been developed to configure the STEP information into XML. This new language lets an information modeler annotate an EXPRESS schema for the purpose of developing an XML Schema. The advantage of this configuration language is that it can produce easy-to-read STEP XML data. The language also enables an information modeler or application developer to tailor the description of the XML for

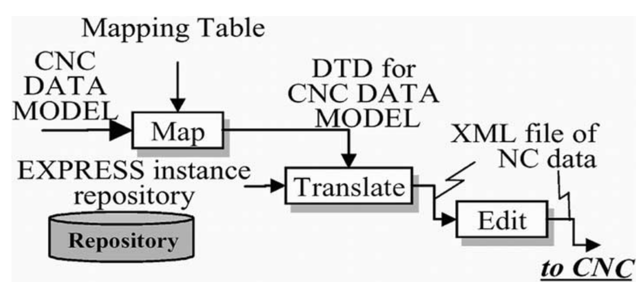

Fig. 8. Generation of XML-based STEP-NC data [63].

an application either to make the data easier to read or easier to process.

There are two different ways of binding EXPRESS semantics to XML, early binding and late binding. In an early binding application, specific tag names are used to correspond directly to their counterparts in EXPRESS. In a late binding, the named components of the XML vocabulary do not directly correspond to the EXPRESS names. Instead, it specifies an EXPRESS name in the data either as an XML attribute value or an element content [62].

There have already been some systems developed with a Web-enabled nature [63], [64]. Fig. 8 shows the generation process of XML-based STEP-NC data from an EXPRESS data model [63]. Manufacturing data based on the CNC data model consist of EXPRESS schema and EXPRESS repositories.

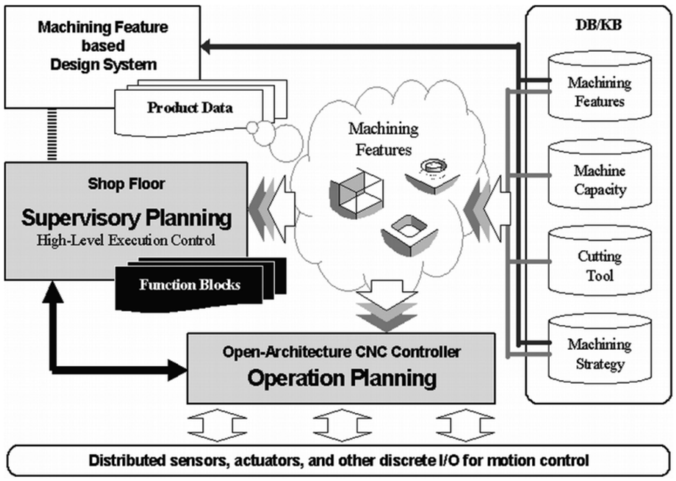

Fig. 9. Data sharing through function blocks and machining features [61].

Using mapping rules, EXPRESS schema is converted to an XML DTD.

A STEP-compliant system framework for collaborative CAPP has been proposed [64]. The framework adopts a three-tiered, Web-based network architecture to provide an open structure for the system. This architecture provides convenient ways in exchanging STEP-NC data in STEP Part 21 and/or Part 28 file format through the Internet. An XML-based STEP-NC milling system was developed in the Seoul National University [65]. The system contains four modules: XML data input module, interpreter, tool-path generator, and motion-control board.

Along the CAD/CAPP/CAM/CNC chain, function blocks play an important role between CAPP and CNC. They are used as information and function carriers with process plans embedded and bridge the gap between CNC machines and shopfloor execution control. Today, more CNC machines are linked to a network, making distributed and interoperable manufacturing possible. In a distributed environment, function blocks are transmitted through network to remote machines whereas run-time machining status can be monitored by triggering the function block-embedded monitoring algorithms. Fig. 9 illustrates one scenario of data sharing enabled by function blocks and machining features. In this case, the tasks of process planning can be divided into two groups and accomplished at two different levels: shop-level supervisory planning and controller-level operation planning. The former focuses on product data analysis, machining feature decomposition, setup planning, machining process sequencing, and fixture and machine selection. The latter considers the detailed Workingsteps for each machining operations, including cutting tool selection, cutting condition assignment, tool-path planning, and control code generation. From design to NC machining, machining features are used for information retrieval, data exchange, and decision-making support at different levels. At the supervisory planning level, the system outputs are a set of function blocks with machining data embedded. They are downloaded to appropriate controllers for low-level operation planning and execution. At the operation planning level, the function blocks are finalized by assigning actual tool data and cutting conditions. As a result, the resultant function blocks are able to describe the detailed operations for the corresponding machining features fabrication.

The very first applications using function blocks for process planning and NC machining [47], [48], [55], [60], [61] are in the area of distributed process planning. A prototype system is being developed at the National Research Council of Canada. It is capable of the machining sequence generation and function block design. More functions, such as machining process monitoring and execution control, are reported in [60].

F. Modularity, Reusability, and Openness

For STEP-NC, these issues are addressed at two different levels. At a macro level, ISO has embarked on an effort to alter the current status of STEP architecture to give APs more interoperability [66]. This is driven by the high-level industry requirements of: a) allowing implementation of a combination of multiple APs or extension of AP implementations with additional capabilities, b) enabling application software reuse, c) eliminating duplication and repeated documentation of the same requirements in different APs, and d) reusing data generated by an implementation of one or more APs, by an implementation of one or more different APs—AP interoperability.

The primary change introduced by the new modular architecture is the explicit harmonization of common information requirements. Rather than relying on harmonization occurring as a byproduct of consistent interpretation across APs, module requirements are first harmonized across domains and the resulting mappings are standardized in smaller packages, known as application modules (AMs). AMs are then reused by other AMs and ultimately in APs [66].

At the micro level, STEP-NC is taking advantage of the existing features of modularity in STEP to address the machining data interoperability issue. The integrated resources that are used to map STEP-NC ARM to AIM were designed with several principles in mind, primarily reusability and extensibility. The integrated resources are a single, consistent, semantically nonredundant model. The model is partitioned into small schemas, each with a unique functionality or purpose. This partitioning facilitates management of the model, enables data reusability, and isolates specific functionalities. In fact, all application protocols are constructed by users from a library of modules that each implements a unit of functionality. These units of functionality are developed by the STEP modelers for each AP, constituting an ARM.

Function blocks, being an IEC standard, follow the object-oriented design methodology. Within the context of process planning, each function block represents an atomic process plan corresponding to the fabrication of a machining feature. Machining features can be assembled to represent a product design. Likewise, function blocks can be reused to form different process plans for different machining tasks. Following the specifications [21], a process planner can modify the modular design of a basic function block or reconnect a subset of basic function blocks to form a composite function block. In terms of its application, a function block not only can represent a partial process plan but also can be used to define an auxiliary function, such as data

TABLE I
ISO 14649 CONFORMANCE CLASSES

Conformance Class	Characteristics of the Conformance Class
1	Minimum set of curve geometry and minimum set of manufacturing data
2	Class 1 plus full set of manufacturing data, especially manufacturing features
3	Class 2 plus minimum set of surface geometry
4	Class 3 plus full curve and surface geometry
5	Class 3 plus topological information
6	Class 4 plus topological information

collection, process monitoring, message forwarding, or fault detection in the form of service interface function blocks.

G. Scalability, Tailor-Ability and Extensibility

Similar to other STEP APs, STEP-NC has been developed with scalability, tailor-ability, and extensibility in mind. This is realized through conformance classes (CC) associated with the standard. CCs are subsets, or tailored versions, of an AP that can be implemented "meaningfully" and independently within that application domain without having to implement all aspects of an AP at the outset. After the initial CCs have been implemented, the implementer can progress, or scale up, to a more complex CC.

There are six classes of conformance (Table I) in STEP-NC ARM (ISO 14 649). They have been carefully defined so that early CC can be easily extended to arrive at a later CC. Similarly, STEP AP 238 defines four CCs: 1) CNC-independent tool paths (CC1); 2) intelligent setup (CC2); 3) conditional programming (CC3); and 4) generative programming (CC4). The current R&D work focuses on CC1.

Compared with STEP-NC, each basic function block is equivalent to a schema with a unique functionality or purpose. Its structure design and algorithm implementation are machine independent. However, its behavior is resource driven. Once a resource is specified, the internal algorithms will find the optimal parameters or solutions to the task for the given resource. Because of its openness, the internal algorithms of a function block can be tailored or extended to suit a new requirement. Its proper combination can meet the requirements of a new industrial application.

H. Portability Among Resources (Machines)

Part program portability has been sought after by CNC manufacturers for many years. This can be achieved through STEP-NC that aims to manufacture parts on different CNC machines. STEP-NC can define a generic data model, which does not require some specific data, such as machine-tool-specific tool paths. Computation for speed and feed data can be left until the very last stage of manufacturing because the design tolerances and surface finishes required for the product are in the AP-224 and AP-203 data and can be connected to a STEP-NC program. Any system that currently produces G-codes can produce STEP-NC data as long as it can understand and work with manufacturing features. Unigraphics, CATIA and Pro/Engineer are all examples of such systems. This allows the STEP-NC programs to be more portable. It also helps save time for NC programming. Contrary to ISO 6983, STEP-NC only allows the use of standard syntax and semantics as outlined in the standards. This avoids the problem existing today with G-code, that for each particular improvement in a given controller, the builder writes an extension for a new subset of instructions in G-code.

Tool-path portability has already been achieved to a certain extent by the joint Boeing/NIST work validating and evaluating AP-238 CC1 for five-axis machining at the end of 2004. The validation proved that five-axis AP-238 CC1 programs with transmission control protocol (TCP) as opposed to axis movement data pertaining to G-code, are portable. During the validation, Boeing used CATIA Version 5 to model the NAS 979 part [67]. It then produced AP238 CC1 tool paths that are translated into CNC vendor-specific TCP programs. Boeing has already successfully demonstrated the ability to produce and run identical TCP programs on multiple five-axis CNCs at Boeing/Tulsa. The OMAC User Group [12] has also taken the lead to spearhead the STEP-NC portability test with participation of some major industrial partners, CAD/CAM vendors, and research institutes. In the OMAC STEP-NC Working Group Meeting [68] held on February 3, 2005, in Orlando, FL, demonstrations were carried out to show how STEP-NC information could support portable machining on five-axis machining centers.

Function blocks are brought to NC machining as a generic CNC controller language [47]. They are generated through distributed process planning with a machine neutral process plan. Such a machine neutral process plan is portable in the form of function blocks to various machine tools with similar capability, where machine-specific data (tool ID, cutting parameters, tool paths, etc.) are generated. In case of a machine failure, the same function blocks can be redirected to another machine for local optimization and execution. According to the distributed process planning (DPP) concept [47], [48], the traditional CAM functionality is decomposed and realized by the shopfloor level supervisory planning and controller level operation planning (Fig. 9). Compared with STEP-NC, function blocks move one step forward in NC machining while maintaining the portability among resources. It is of the authors' opinion that function blocks fare better when it comes to interfacing with hardware.

V. CONCLUSION

Modern CNC machine tools, though capable in functionalities, lack adaptability, portability, and intelligence. This is due to the fact that a 50-year-old language (G-code) is still employed by these machine tools. NC programs following this format are only meant for execution on a specific machine tool. They cannot be reinterpreted by a CAM system or a shop-floor program (SFP) system for a different machine tool. Automatic generation of a 100% optimized NC program is not possible as design information and know-how about the machine tools and materials is represented in different formats and on different databases.

STEP-NC and function blocks are the two standards that seem to hold the key for a truly interoperable CNC environment. The main contribution of this paper has been the provision of an in-depth and parallel technological review of the two technologies from the viewpoint of interoperable manufacturing. Some of the early prototyping work is also presented underlining a potential fusion of the technologies in search for interoperability. To a varying degree, both standards, though unrelated, support interoperable machining. STEP-NC completes the entire product development chain by offering a STEP-compliant link between CAPP and CNC. It can therefore supply CNCs with a complete product model. STEP-NC is good at supporting bidirectional information flow in CAD/CAM, data sharing over the Internet, use of the feature-based machining concept, modularity and reusability, and portability among resources. Function blocks, on the other hand, provide a useful tool for developing interoperable CNC controllers and the control strategies. This is because a function block can be viewed as a fundamental functional and executable unit. It is not suitable to model a design but machining information. Function blocks can be designed to match individual machining features and have needed algorithms and data embedded to decide the best cutting conditions and tool path once a machine and tool are selected. This makes function blocks best suited for CNC controls. They have an edge over STEP-NC in supporting intelligent and autonomous CNC, use of feature-based machining concept, portability among resources, and dynamism and adaptiveness. In short, STEP-NC can be viewed as a "job setter" (providing all of the necessary information), whereas function blocks can be viewed as a "job doer" (executing the machining commands) for interoperable manufacturing. To this end, STEP-NC and function blocks can work hand in hand in supporting interoperable manufacturing. Despite the short history of STEP-NC and function blocks, there is no shortage of research projects utilizing the standards for interoperable manufacturing. However, up to now, there is no system combining both STEP-NC and function blocks. There also seems to be some resistance in uptaking the technologies, mostly attributed to the widespread and dominated use of G-code on CNC controllers. Customers' demands on a CNC machine tool for it to be truly interoperable, intelligent, and autonomous will continue to push the technology to mature to a level at which commercialization kicks in and takes over. It is of the authors' humble opinion that STEP-NC and the function block need to make headway in separate domains for them to eventually and collectively provide viable solutions to the manufacturing industry. STEP-NC should readily appeal to CAD/CAM vendors more than any other parties as STEP (i.e., ISO 10 303-203 and 214) is already a regular import/export option in most of the CAD/CAM systems. On the other hand, the motion-control suppliers for CNC OEMs may take interest in function blocks as they are hardware related and modular in nature. This being said, prototyping work on combining STEP-NC with function blocks must not be delayed as it is abundantly clear that they both promise a multitude of benefits over the current CAPP/CAM/CNC regime. The future research focus is likely to center around using STEP-NC as the information provider and function blocks as the enabler for interoperable manufacturing.

Acknowledgment

The authors would like to thank the U.S. National Institute of Standards and Technology (NIST) for sponsoring the Guest Research program which provided the authors with the opportunity of drawing upon the expertise from a number of researchers at NIST. Special thanks also to F. Proctor, T. Kramer, J. Michaloski, A. Domez, and J. Soons in the Manufacturing Engineering Laboratory of NIST. The authors are also grateful to the STEP-NC and Function Block committees, in general, for the provisions of the draft standards and other relevant information.

References

[1] *ISO 6983-1:1982 Numerical Control of Machines—Program Format and Definition of Address Words—Part 1: Data Format for Positioning, Line Motion and Contouring Control Systems*, 1982.

[2] S. Brummermeirier and S. Martin, Interoperability Cost Analysis of the US Automotive Supply Chain, Research Triangle Inst., Research Triangle Park, NC, 1999.

[3] Advanced Manufacturing Technology Research Institute, Agile Reconfigurable Manufacturing Machinery Systems—ARMMS, Project Summary for Brite-Euram Nr: BRRT-CT98-5080, Jan. 1, 1999–Dec. 31, 2001.

[4] F. M. Asl and A. G. Ulsoy, "Stochastic optimal capacity management in reconfigurable manufacturing systems," *CIRP Ann.—Manufact. Technol.*, vol. 52, no. 1, pp. 371–374, 2003.

[5] M. Bruccoleri, M. Amico, and G. Perrone, "Distributed intelligent control of exceptions in reconfigurable manufacturing systems," *Int. J. Prod. Res.*, vol. 41, no. 7, pp. 1393–1412, 2003.

[6] J. Mun, K. Ryu, and M. Jung, "Self-reconfigurable software architecture: design and implementation," in *Proc. 33rd Int. Conf. Computers Industrial Engineering*, Jeju, Korea, Mar. 25–27, 2004, CIE569.

[7] K. Ryu and M. Jung, "Agent-based fractal architecture and modeling for developing distributed manufacturing systems," *Int. J. Prod. Res.*, vol. 41, no. 17, pp. 4233–4255, 2003.

[8] C. Yuan and P. Ferreira, "An integrated rapid prototyping environment for reconfigurable manufacturing systems," in *Proc. 2003 ASME Int. Mechan. Eng. Congr.*, Washington, D.C., Nov. 15–21, 2003.

[9] Manufacturing Data Systems Inc.. (2005, Mar.) OpenCNC Broch.. [Online] Available: http://www.mdsi2.com/Solutions/CNC_Controls/Brochure/OpenCNCbrochure.pdf.

[10] P. Lutz and W. Sperling *et al.*, "OSACA—the vendor neutral control architecture. In: 'facilitating deployment of information and communications technologies for competitive manufacturing'," in *Proc. Eur. Conf. Integr. Manufact.*, D. Fichtner *et al.*, Eds., Dresden, Germany, 1997.

[11] OSEC (Open System Environment for Controller). (2005, Aug.) Open System Environment for Controller Architecture Overview. [Online] Available: ttp://osec.skk-inc.co.jp/draft1.0/welcome.

[12] D. Brown. (2005, Mar.) Open Modular Architecture Controls: OMAC-HMI, OSACA, JOP-Standard CNC Data Type Analysis. ROY-G-BIV Corp, Bingen, WA. [Online] Available: http://www.omac.org/wgs/MachTool/HMI-API/standards_compare.pdf.

[13] S. C. Feng and E. Y. Song, "A manufacturing process information model for deign and process planning integration," *J. Manufact. Syst.*, vol. 22, no. 1, pp. 1–15, 2003.

[14] F. L. Zhao, S. K. Tso, and P. S. Y. Wu, "A cooperative agent modeling approach for process planning," *Comput. Ind.*, vol. 41, pp. 83–97, 2000.

[15] O. Lopez-Ortega, "Java application based on an EXPRESS model for sharing flexible manufacturing resources data," in *Proc. IEEE Symp. Emerging Technologies Factory Automation*, vol. 2, 2001, pp. 3–11.

[16] *ISO 14 649-1: 2003, Data Model for Computerized Numerical Controllers: Part 1 Overview and Fundamental Principles*, 2003.

[17] *ISO 14 649-10: 2003, Data Model for Computerized Numerical Controllers: Part 10—General Process Data*, 2003.

[18] *ISO 14 649-11: 2003, Data Model for Computerized Numerical Controllers: Part 11—Process Data for Milling*, 2003.

[19] *ISO 14 649-111: 2001 Data Model for Computerized Numerical Controllers: Part 111—Tools for Milling*, 2001.

[20] *ISO/DIS 10 303-238: 2003, Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 238: Application Protocols: Application Interpreted Model for Computerized Numerical Controllers*, 2003.

[21] *IEC TC65/WG6: 1999. Function Blocks for Industrial Process Measurement and Control Systems (Part-1: Architecture)*, 1999.

[22] B. Zhou, L. Wang, and D. Norrie, "Design of distributed real-time control agents for intelligent manufacturing systems," in *Proc. 2nd Int. Workshop on Intelligent Manufacturing Systems*, Leuven, Belgium, Sep. 22–24, 1999, pp. 237–244.

[23] *ISO 10 303-21: 1994, Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure*, 1994.

[24] *ISO 10 303-22: 1998. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 22: Implementation Methods: Standard Data Access Interface*, 1998.

[25] *ISO 10 303-23: 2000. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 23: C++ Language Binding to the Standard Data Access Interface*, 2000.

[26] *ISO 10 303-24: 2001. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 24: C Language Binding of Standard Data Access Interface*, 2001.

[27] *ISO 10 303-27: 2000. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 27: Java Programming Language Binding to the Standard Data Access Interface With Internet/Intranet Extensions*, 2000.

[28] *ISO/CD TS 10303-28 (Edition 1): 2002. Product Data Representation and Exchange: Implementation Methods: EXPRESS to XML Binding, Draft Technical Specification, ISO TC184/SC4/WG11 N169, 2002-02-14*, 2002.

[29] *ISO/TS 10 303-28 (Edition 2): 2004. ISO ISO/WD 10 303-28 (Edition 2), Product Data Representation and Exchange: Implementation Methods: XML Schema Governed Representation of EXPRESS Schema Governed Data, TC184/SC4/WG11 N223, 2004-02-17*, 2004.

[30] *ISO/WD 10 303-29: 2004. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 29: Implementation Methods: XML Encoding of the Exchange Structure of Product Data Defined by An Application Protocol*, 2004.

[31] Object Management Group. (2003, Mar.) OMG Unified Modeling Language Specification. Version 1.5. [Online] Available: http://www.omg.org/technology/documents/formal/uml.htm.

[32] *STEP-NC Newslett. Issue 3*, Nov. 2000.

[33] X. W. Xu and Q. He, "Striving for a total integration of CAD, CAPP, CAM and CNC," *Robot. Comput. Integr. Manufact.*, vol. 20, pp. 101–109, 2004.

[34] J. Wolf, "Requirements in NC machining and use cases for STEP-NC—analysis of ISO 14 649 (ARM) and AP 238 (AIM)," in *ISO T24 STEP-Manufacturing Meeting*, South Diego, CA, Mar. 2003.

[35] A. B. Feeney, T. Kramer, F. Proctor, M. Hardwick, and D. Loffredo, "STEP-NC implementation—ARM or AIM?," in *ISO T24 STEP-Manufacturing Meeting*, South Diego, CA, Mar. 2003.

[36] X. W. Xu, H. Wang, J. Mao, S. T. Newman, T. R. Kramer, F. M. Proctor, and J. L. Michaloski, "STEP—Compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration," *Int. J. Prod. Res.*, vol. 43, no. 17/1, pp. 3703–3743, 2005.

[37] "Tech. Rep. 3 of IMS Project (97006) STEP-Compliant Data Interface for Numerical Controls (STEP-NC)," IMS STEP-NC Consortium, Jan. 1–June 31 2003.

[38] W. Maeder, V. K. Nguyen, J. Richard, and J. Stark, "Standardization of the manufacturing process: the IMS STEP-NC project," in *Proc. INational Network of Competence on Integrated Production Logistics) Workshop*, Saas Fee, Switzerland, Sep. 2002, pp. 5.5/1–5.5/3.

[39] S. Storr and S. Heusinger, STEP-NC—Basis of a CAD/NC Process Chain—The STEP-NC Process Model for Turning.

[40] (2005, Mar.) ST-Plan. [Online] Available: http://www.steptools.com/products/stplan/.

[41] "STEP-NC demonstrated at JPL," *Manufacturing Engineering*, vol. 130, no. 3, pp. 34–35, Mar. 2003.

[42] S. H. Suh, D. H. Chung, J. H. Lee, J. H. Cho, H. D. Hong, and H. S. Lee, "Developing an integrated STEP-compliant CNC prototype," *J. Manufact. Syst., SME Trans.*, vol. 21, no. 5, pp. 350–362, 2003.

[43] S. H. Suh, B. E. Lee, D. H. Chung, and U. S. Cheon, "Architecture and implementation of a shop-floor programming system for STEP-compliant CNC," *Comput.-Aided Design*, vol. 35, pp. 1069–1083, 2003.

[44] S. T. Newman, "Integrated CAD/CAM/CNC manufacture for the 21st century," in *Proc. 14th International Conference on Flexible Automation and Intelligent Manufacturing* . Toronto, ON, Canada, Jul. 12–14, 2004.

[45] S. T. Newman, R. D. Allen, and R. S. U. Rosso Jr., "CADCAM solutions for STEP-compliant CNC manufacture," *Int. J. Comput. Integr. Manufact.*, vol. 16, no. 7–8, pp. 590–597, 2003.

[46] X. W. Xu, "Development of a G-code free, STEP-compliant CNC lathe," in *Proc. Int. Mechan. Eng. Congr. Expo. ASME Winter Conf.*, Anaheim, CA, Nov. 13–19, 2004, IMECE2004-60346, CIE-2, pp. 1–5.

[47] L. Wang, "DPP: a distributed process planning approach using function block," in *Proc. ASME 2002 Design Engineering Technical Conf.*, Montreal, QC, Canada, Sep. 29–Oct. 2 2002, DETC2002/DFM-34 194.

[48] L. Wang, H.-Y. Feng, and N. Cai, "Architecture design for distributed process planning," *J. Manufact. Syst.*, vol. 22, no. 2, pp. 99–115, 2003.

[49] B. L. M. Goldstein, S. J. Kemmerer, and C. H. Parks, A Brief History of Early Product Data Exchange Standards, Nat. Inst. Standards Technol., Gaithersburg, MD, Sep. 2, 1998.

[50] STEP: The Grand Experience, S. J. Kemmerer, Ed., Manufact. Eng. Lab., Nat. Inst. Standards Technol., Gaithersburg, MD, July 1999.

[51] M. Weck, "STEP-NC: A new interface closing the gap between planning and shopfloor," in STEP-NC Workshop, Werkzeugmaschinenlabor (WZL), Rheinisch-Westfdlische Technische Hochschule (RWTH), Aachen, Germany, Feb. 2003.

[52] M. Weck and J. Wolf, ISO 14 649 Provides Information for Sophisticated and Flexible Numerically Controlled Production, 2002.

[53] M. Weck, J. Wolf, and D. Kiritsis, "STEP-NC—the STEP compliant NC programming interface evaluation and improvement of the modern interface," in *Proc. ISM Project Forum*, Genf, Switzerland, Oct. 2001.

[54] D. Kiritsis, "STEP-NC - integrating shop floor into industrial data flow for the enabling of intelligent near to process functions," in *Proc. IMS Project Forum*, Ascona, Switzerland, Oct. 9, 2001.

[55] L. Wang and W. Shen, "DPP: an agent-based approach for distributed process planning," *J. Intell. Manufact.*, vol. 14, no. 5, pp. 429–439, 2003.

[56] F. Zhang, Y. F. Zhang, and A. Y. C. Nee, "Using genetic algorithms in process planning for job shop machining," *IEEE Trans. Evol. Comput.*, vol. 1, no. 4, pp. 278–289, Nov. 1997.

[57] G. H. Ma, Y. F. Zhang, and A. Y. C. Nee, "A simulated annealing-based optimization algorithm for process planning," *Int. J. Prod. Res.*, vol. 38, no. 12, pp. 2671–2687, 2000.

[58] S. H. Suh, J. H. Cho, and H. D. Hong, "On the architecture of intelligent STEP-compliant CNC," *Int. J. Comput. Integr. Manufact.*, vol. 15, no. 2, pp. 168–177, 2002.

[59] S. H. Suh and S. U. Cheon, "A framework for an intelligent CNC and data model," *Int. J. Adv. Manufact. Technol.*, vol. 19, no. 10, pp. 727–735, 2002.

[60] L. Wang, W. Jin, and H.-Y. Feng, "Design and event-driven control of function blocks for adaptive process plan execution," *Trans. North Amer. Manufact. Res. Inst. SME*, vol. 33, pp. 343–350, 2005.

[61] L. Wang, Z. Liu, W. Shen, and S. Lang, "Function-block enabled job shop planning and control with uncertainty," in *Proc. ASME Int. Mechan. Eng. Congr. Expo.*, Anaheim, CA, Nov. 13–19, 2004, IMECE2004-59 279.

[62] J. Lubell and S. Frechette, "XML representation of STEP schemas and data," *Trans. ASME: J. Comput. Inform. Sci. Eng.*, vol. 2, no. 2, pp. 69–71, Mar. 2002.

[63] T. Kishinami, F. Tanaka, and Y. Sato, "XML-based high level NC data modeling for manufacturing data management," in *Proc. ISO Charleston Meeting*, Charleston, SC, Oct. 19, 2000.

[64] X. W. Xu and J. Mao, "A STEP-compliant collaborative product development system," in *Proc. 33rd Int. Conf. Computers Industrial Engineering*, Jeju, Korea, Mar. 25–27, 2004, CIE598.

[65] W. Lee and Y.-B. Bang, "Design and implementation of an ISO14649-compliant CNC milling machine," *Int. J. Prod. Res.*, vol. 41, no. 3, pp. 3007–3017, 2003.

[66] A. B. Feeney, "The STEP modular architecture," *Trans. ASME: J. Comput. Inform. Sci. Eng.*, vol. 2, pp. 132–135, Jun. 2002.

[67] NAS 979, Uniform Cutting Tests—NAS National Aerospace Standard Series: Metal Cutting Equipment Specifications, National Aerospace Standard, National Standards Association, Washington, D.C.

[68] (2005, Mar.) STEP-NC pilot projects. OMAC Users Group Meeting. [Online] Available: http://www.isd.mel.nist.gov/projects/stepnc.

Xun W. Xu received the B.Sc. and M.Sc. degrees from Shenyang Jianzhu University and Dalian University of Technology, Dalian, China, in 1982 and 1988, respectively, and the Ph.D. degree from the Department of Mechanical Engineering, University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K.

Currently, he is a Senior Lecturer in the Department of Mechanical Engineering, University of Auckland, Auckland, New Zealand. He heads the Manufacturing Systems Laboratory and the CAD/CAM Laboratory in the University of Auckland. His main interests are CAD/CAPP/CAM, feature technology, STEP-NC, interoperable manufacturing, and product life-cycle management.

Dr. Xu is a member of the American Society of Mechanical Engineers (ASME) and the Institution of Professional Engineers New Zealand (IPENZ).

Lihui Wang received the M.E.Sc. and Ph.D. degrees from Kobe University, Kobe, Japan, in 1993 and 1990, respectively, and the B.Sc. degree from the Academy of Arts and Design, Beijing, China, in 1982.

Currently, he is a Senior Research Officer of Integrated Manufacturing Technologies Institute, National Research Council (NRC), London, ON, Canada. He was an Assistant Professor with Kobe University and then with Toyohashi University of Technology, Toyohashi, Japan. He is also an Adjunct Professor with the University of Western Ontario, London, ON, Canada. His research interests are manufacturing process planning, planning and scheduling integration, agent technology, real-time monitoring and control, and Java and web-based systems.

Dr. Wang is a senior member of the Society of Manufacturing Engineers (SME) and a member of the American Society of Mechanical Engineers (ASME).

Yiming Rong received the B.Sc. degree from Harbin University of Science and Technology, Harbin, China, in 1981; M.S. degrees from Tsinghua University, Beijing, China, and the University of Wisconsin, Madison, in 1984 and 1987, respectively; and the Ph.D. degree from the University of Kentucky, Lexington, in 1989.

Currently, he is the Higgins Professor of Mechanical Engineering and the Director of the Computer-Aided Manufacturing Laboratory (CAM Lab.), Department of Mechanical Engineering, Worcester Polytechnic Institute (WPI), Worcester, MA. He was a Faculty Member with Southern Illinois University, Carbondale, for eight years before joining WPI in 1998. His research areas are computer-aided manufacturing, including manufacturing systems, manufacturing processes, and computer-aided fixture design (CAFD). The research on CAFD has been recognized nationally and internationally.

Dr. Rong is a fellow of the American Society of Mechanical Engineers (ASME) and a member of the Society of Manufacturing Engineers (SME) and the American Society for Engineering Education (ASEE).