

---

## **STEPNCMillUoA: a CNC system based on STEP-NC and Function Block architecture**

---

**Mohamad Minhat and Xun Xu\***

Department of Mechanical Engineering,  
The University of Auckland,  
Private Bag 92019, Auckland Mail Centre,  
Auckland 1142, New Zealand  
E-mail: mmin018@aucklanduni.ac.nz  
E-mail: x.xu@auckland.ac.nz  
\*Corresponding author

**Valeriy Vyatkin**

Department of Electrical and Computer Engineering,  
The University of Auckland,  
Private Bag 92019, Auckland Mail Centre,  
Auckland 1142, New Zealand  
E-mail: v.vyatkin@auckland.ac.nz

**Abstract:** STEPNCMillUoA is the prototype of a new CNC system that utilises the STEP-NC data model and IEC 61499. STEP-NC provides a high-level data model and enables feature-based machining whereas the enabled layered Function Block architecture simplifies the design of the CNC controller. The architecture layers are responsible for data processing, storage and execution. The object-oriented Model-View-Control design pattern supports the system architecture and the design framework, in which simulation of the machining becomes natural and inherent part of the design process, with seamless transition from simulation to actual machining. This system possesses interoperability, portability, re-configurability and distribution characteristics. The system was tested through simulation and actual machining.

**Keywords:** CNC; machine tools; STEP-NC; IEC 61499 standard; FB; function blocks; architecture; feature-based; interoperability; portability; re-configurability.

**Reference** to this paper should be made as follows: Minhat, M., Xu, X. and Vyatkin, V. (2009) 'STEPNCMillUoA: a CNC system based on STEP-NC and Function Block architecture', *Int. J. Mechatronics and Manufacturing Systems*, Vol. 2, Nos. 1/2, pp.3–19.

**Biographical notes:** Mohamad Minhat obtained his Masters of Science from Coventry University, UK in 2003 and a Bachelor in Mechanical and Manufacturing Engineering from UWCC (then Cardiff University), UK in 1994. He is a Lecturer at the UTeM, Malaysia since 2001. He has experiences in the academic field as well as the industries. He was with Aero-Composites Technologies (CTRM, Malaysia, 2000) and Samsung Display International (Malaysia, 1995~2000). He is now a Malaysian Government perusing his doctoral research at the University of Auckland.

Xun Xu received a PhD from the University of Manchester (then UMIST), UK in 1996. He is now an Associate Professor at the Department of Mechanical Engineering, the University of Auckland, New Zealand. He is a member of ASME and IPENZ. He has also been actively engaged in various industrial consultancy work. He heads the Manufacturing Systems Laboratory in the University of Auckland. He has published numerous research monographs and is on the editorial boards for a number of international journals. His main interests lie in the areas of CAD/CAPP/CAM, STEP, and STEP-NC.

Valeriy Vyatkin is a Senior Lecturer at the University of Auckland, New Zealand. His main research interests are software engineering for industrial automation systems, distributed software architectures and methods of formal validation of industrial automation systems, in particular distributed automation and the IEC 61499 standard. He is the author of 120 international publications and of several software tools. He received the PhD and the Dr. Sci Degrees from Taganrog State University of Radio Engineering (TSURE), Taganrog, Russia in 1992 and 1998 respectively, and the Dr. Eng Degree from the Nagoya Institute of Technology, Japan, in 1999.

---

## 1 Introduction

Most of the CNC controllers run off G-codes, the programming language standardised by ISO 6983 (ISO 6983-1, 1982). G-code programs are usually generated by CAM tools, using as input the geometry data from CAD tools. The way G-codes are programmed remains almost unchanged in the last 50 years. So do the bottle-neck problems such as lack of program portability. Different CNC vendors tend to implement different versions of G-codes for different CNCs; this is handled by a proprietary unit called post-processor. In addition to a detailed description of the machine tool and other parameters such as cutting tools and other auxiliary components, the post-processor also needs to know the particular brand and model of a CNC machine. Consequently, a CNC machine can only accept its proprietary G-codes, which hinders flexibility and interoperability of machining information especially within globalised industries.

Recently, a new NC data model based on the Standard for the Exchange of Product (STEP) has been developed to address the problems. This data model is known as STEP-NC standardised by ISO 14649-1 (2003) and ISO 10303-238 (2007). It defines process plans, features, machining sequences, machining methods, tool information and other manufacture related information. STEP-NC is meant to be neutral to different CNC controllers, hence post-processing is not required.

A STEP-NC part program comprises Workingsteps that are effectively sequentially executed machining operations. Each working step has information about its machining operation, manufacturing feature, geometry, machining strategy, machining technology, machine function, and machining tool. Therefore, STEP-NC recommends a rich data format, and offers machine tool controllers high-level data.

The benefits of the STEP-NC data model can be further strengthened if combined with an open-architecture CNC controller, and object-oriented, high-level machining interface or language (Nassehi et al., 2006; Ho et al., 2005; Brecher et al., 2006; Fortin et al., 2004; Peng et al., 2005; Heusinger et al., 2006; Liu and Zhang, 2004).

There has been no shortage of research work in this field in the recent past (Xu et al., 2006; Zhang et al., 2006; Kramer et al., 2006; Mathews, 2002). The research started in Europe and the USA, with participants from manufacturers of all systems related to the data interface (i.e., CAM systems, CNC controls, and machine tools), to the end users and academic institutions. Han et al. (2007) proposed a novel framework based on multi-agents for implementing an intelligent STEP-NC controller. In the development of a PC-based milling machine, Lee et al. (2006) explained and tested a STEP-NC interpreter that was built using XML and the STEP-NC controller concept. Patents concerning STEP-NC controllers and part programming work can also be found (POSTECH Foundation, 2008). In 2005, Boeing and the US National Institute of Standards and Technology (NIST) worked jointly to prove that using STEP-NC for tool centre programming could assist exchange of information between process planning, work preparation, tooling, and other production processes on different machine tools (Venkatesh et al., 2005). Xu (2006) demonstrated a G-code free machining scenario on a STEP-compliant CNC lathe with a more open platform control system, i.e., Compumotor Motion Control (CMC) system. Storr et al. (2002) from the University of Stuttgart developed a software tool implementing planning methodologies for the STEP-NC enabled sequential and parallel automatic machining.

Another standard that has the potential of advancing CNC controllers is IEC 61499 also known as Function Blocks (FB) (IEC 61499, 2005). This standard was developed by the International Electrotechnical Commission (IEC) for the use of software objects (i.e., FB) in distributed Industrial-Process Measurement and Control Systems (dIPMCS). The need for this new standard resulted from several studies and research programs that were started or conducted in the late 1980s and early 1990s of the last century (Zoitl et al., 2007). The IEC 61499 standard can satisfy the new requirements of automation and control systems and production facilities for them to be more flexible, adaptable and reconfigurable (Grabmair et al., 2007). FB have been adopted by IEC to design reusable, interoperable and distributed control applications (Vyatkin, 2007), and the first fully fledged tool support for IEC 61499 has been recently developed by ICS Triplex in their product called ISaGRAF as explained by Chouinard et al. (2007) and Vyatkin and Chouinard (2008).

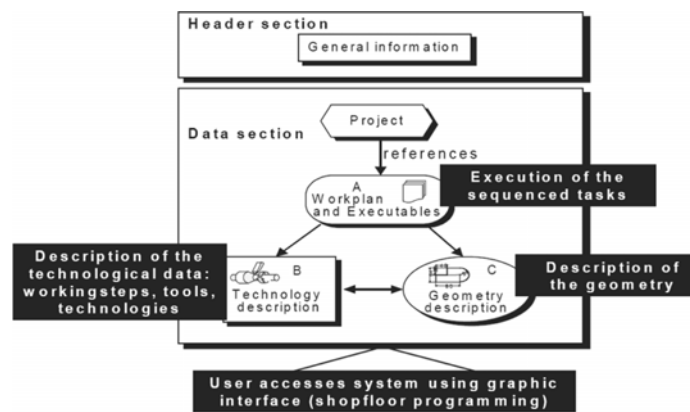
More recently, there are a number of researchers considering a combined use of the two new standards, STEP-NC and FB. The research started in the late 1990s when researchers were trying to increase adaptability, flexibility and interoperability for CNC machining operations. Minhat et al. (2008) demonstrated that the IEC 61499 reference architecture can be effectively used for the development of a new generation CNC controller, offering interoperability, portability, configurability and distribution characteristics. Wang et al. (2007) introduced an adaptable CNC system based on STEP-NC and FB. Xu et al. (2006) discussed the roles that both STEP-NC and FB can play in achieving more interoperable manufacturing. Wang (2002) and Wang et al. (2006) presented a distributed process planning approach using FBs. The primary focus of the research was on the architecture of a new process planning system using FBs as the controller language. It is based on a design-for-machining concept that can seamlessly integrate feature-based design and agent-based planning into the FB-based CNC control.

This paper discusses the two new standards, ISO 14649 and IEC 61499, with a focus on the development of a STEP-NC enabled controller. The controller architecture has been presented and a prototype system based on this architecture has been developed. The development tools used in this research are also discussed.

## 2 ISO 14649 (STEP-NC) and IEC 61499 (Function Block)

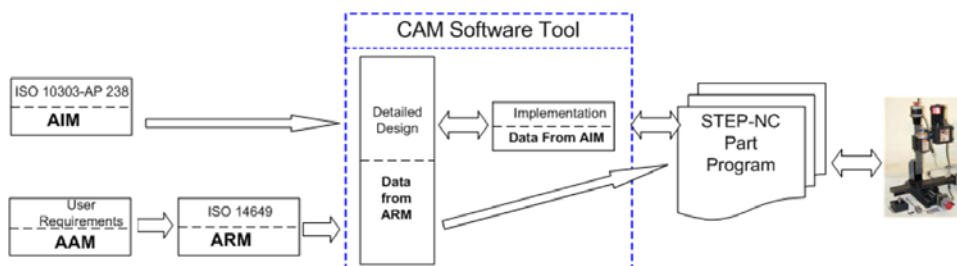
STEP-NC contains a suite of ISO parts such as ISO 14649-1 (2003), ISO 14649-10 (2004), ISO 14649-11 (2003) and ISO 14649-111 (2003). The aim is to enhance data exchange between CAD/CAM and CNC (Xu et al., 2005, 2006). Figure 1 illustrates the structure of a STEP-NC file. The root has a Project containing a main Workplan. Each Workplan may contain other Workplans and/or a series of Workingsteps. Each Workingstep defines a machining operation for a manufacturing feature, for example, rough-mill a pocket. The same machining operation may be used in several Workingsteps and more than one Workingsteps may be necessary to complete a feature. For example, completion of a pocket may require a roughing operation followed by a finishing operation.

**Figure 1** The structure of a STEP-NC file



In fact, STEP-NC can be exploited using two different standards, ISO14649 and ISO10303-AP238. The former is a manufacturing Application Reference Model (ARM), whereas the latter is an Application Interpreted Model (AIM). This means that ISO 14649 abstracts the Application Activity Model (AAM) and mainly describes the requirement data for CNC machining. ISO10303-AP238, on the other hand, maps ARM to generic STEP resources to form an AIM. Both have been used in support of CNC machining (Figure 2).

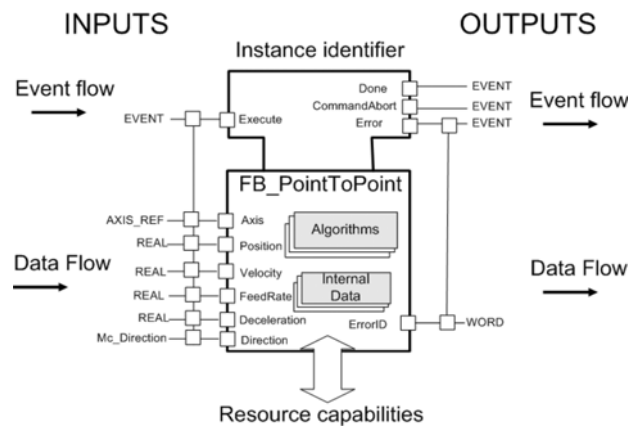
**Figure 2** Two implementation methods (see online version for colours)



A FB can be interpreted as a software unit that encapsulates algorithms which can be designed to behave in a similar way as an electronic device or a circuit and it can

represent a small task in a control plan. It can also encapsulate multiple small control units. The IEC-61499 FB specification is meeting the requirement based on an explicit event driven model and also provides data flow and finite-state automata-based control. There are Basic FB and Composite FB. A Basic FB defines the fundamental functional relationships of events and data. A Composite FB is a combination of several Basic FB which look just like a ‘normal’ FB from the outside. Figure 3 shows an example of FB implementing one-axis, point-to-point movement in NC machining. The upper part shows the event inputs and outputs. The data inputs and outputs are shown at the lower part. The event input ‘Execute’ is associated with all data inputs such as ‘Axis’ (reference to a particular axis), or motion parameters such as ‘Velocity’, or ‘Feed-rate’. The FB is activated on event and the event data association ensures that the relevant data values are updated.

**Figure 3** A basic Function Block



The algorithms can be written in either high-level programming languages, e.g., Java™ or the IEC 61131 languages for programmable controllers, e.g., Ladder Diagrams and Structured Text.

Previous research on FB proved that they 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. They are also suitable for machine level monitoring, shop-floor execution control and CNC control.

These two standards (STEP-NC and FB) are therefore used in this research for the development of a new generation CNC system called STEPNCMillUoA. The remaining part of the paper discusses the architecture of the system and its prototype.

### 3 System's architecture

#### 3.1 Overview

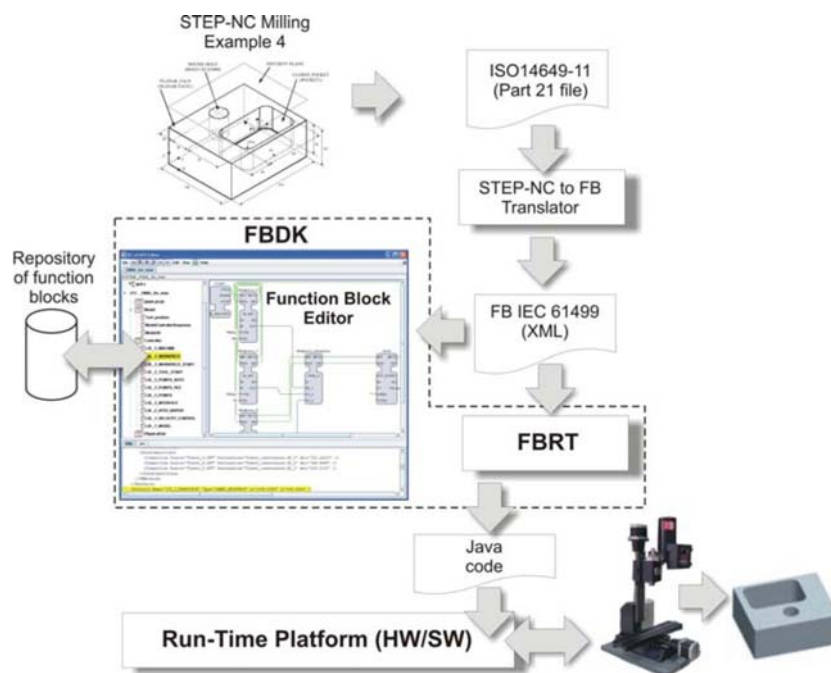
The architecture of the proposed CNC system is presented in Figure 4. It consists of

- STEP-NC data
- STEP-NC to FB translator

- prototype of embedded CNC-FB
- STEP-NC controller.

The design flow starts with the development of a STEP-NC data model describing the part to be manufactured. The STEP-NC input file (Part 21 physical file) contains both design and machining data that are correlated to other necessary information in part programming, such as tooling requirements, feature parameters, feature geometry, feature tolerances, strategy and technology information.

**Figure 4** Proposed architecture and data flows of the STEPNCMillUoA (see online version for colours)



Then the STEP-NC to FB translator interprets the STEP-NC data and generates the equivalent FB model, which can be manually modified using the FB Editor. The FB model is structured according to the object-oriented MVC design pattern (Christensen, 2000). The generation of tool path is later carried out by the STEP-NC controller, which is capable of direct execution of FB. The key software components used to implement the architecture are discussed in the following subsections.

### 3.2 STEP-NC to FB translator

The STEP-NC to FB translator takes STEP-NC files as the input that gives a description of the structural properties defining all the entities represented in a CNC milling application. The STEP-NC data model is programmed using the EXPRESS language. EXPRESS data models do not define any implementation methods. Hence, implementation method, i.e., STEP Part 21 (ISO 10303-21, 1994) is used. In this research, a set of software tools and programming libraries developed by ST-Developer

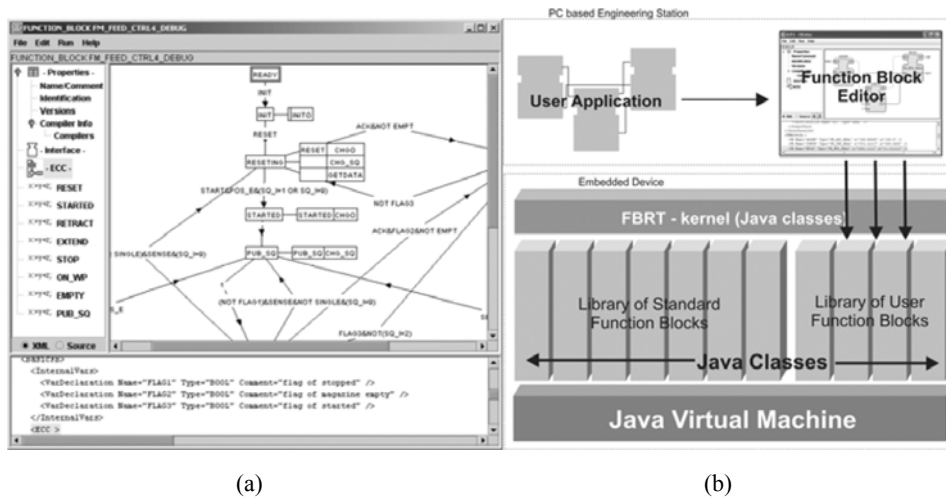
(STEP Tools, Inc., 2008) are used. These tools and library functions can work directly with EXPRESS information models and EXPRESS-defined data sets. The software also provides EXPRESS schema tools in form of C++. This is done by producing C++ source code, which can be used for objects (with these data structures and libraries) to read, write, create, search, and manipulate any STEP and STEP-NC data. The user interface developed helps manage and transfer the necessary data for generation of the NC subroutines for execution.

### 3.3 Function Block implementation software

FB are executable specifications; they can be directly executed if the corresponding software support is provided. Essential are a tool for editing FB networks, a compiler to executable code and some run-time support, such as standard libraries, also known as run-time platforms. The FB concept matches well with engineering methodologies that help to develop efficiently complex automation systems including visualisation, simulation models and Human-Machine Interface (HMI) along with usual control and communication.

In this research, we used FB Development Kit (FBDK, 2007), a Java-based tool, originally developed by the Advanced Technology Division of Rockwell Automation (currently supported and distributed by HOLOBLOC, Inc., USA). It contains a graphical FB Editor (Figure 5(a)) – an integrated development environment that supports graphical development of FB and systems and their translation to Java™ classes. The FB Run-Time (FBRT) (Figure 5(b)) complements the editor with the ability to execute the compiled FBs. The latter can be executed on any computer that supports the Java™ Virtual Machine.

**Figure 5** (a) Function Block editor and (b) Java-based run-time environment FBRT



Source: Vyatkin (2007)

Although Java-based solutions are notorious for poor real-time performance, it was demonstrated by Lastra et al. (2005), that after certain optimisations the Java-based control is up to speed for various performance-demanding applications.

Most importantly, the implementation-independent nature of IEC 61499 guarantees an easy transition to a more real-time capable platform, such as ISaGRAF or FORTE (2008).

## 4 System workflow

The system application is named as STEPNCMillUoA and it is more than just a program. It is a distributed system composed of three main devices and one physical output device with several resources, which are all organised internally. The resource is populated by the network of FBs that are software components calling each other and passing data between each other.

### 4.1 Translating STEP-NC data to FBs

STEP-NC data are object-oriented. Commonly used objects are Workplan, Workingstep, Machining feature, etc., STEP-NC data are also of a hierarchical structure with a predominantly top-down inheritance. In this research, STEP-NC data are represented as functional units that correspond to layered FB architecture. In this architecture, each layer is designed to utilise the services of the lower one. Communications between the layers are implemented by the communication FBs *Publish* and *Subscribe*.

For example, machining features are implemented by a library of corresponding FBs that parameterise machining features and machining data. Subsequent FBs generate the tool paths required for the respective features. Other FBs are used for managing features and shape coordinates to accept or collect data entered either manually by user, or automatically from other data sources.

In particular, the very first set of data translated from a STEP-NC data model are related to set-up initialisation, which include:

- Machine Coordinate System (MCS) (the maximum travel displacement of each axis)
- workpiece (dimensions)
- workpiece coordinate system (workpiece origin)
- path coordinate system (tool start point-home positions).

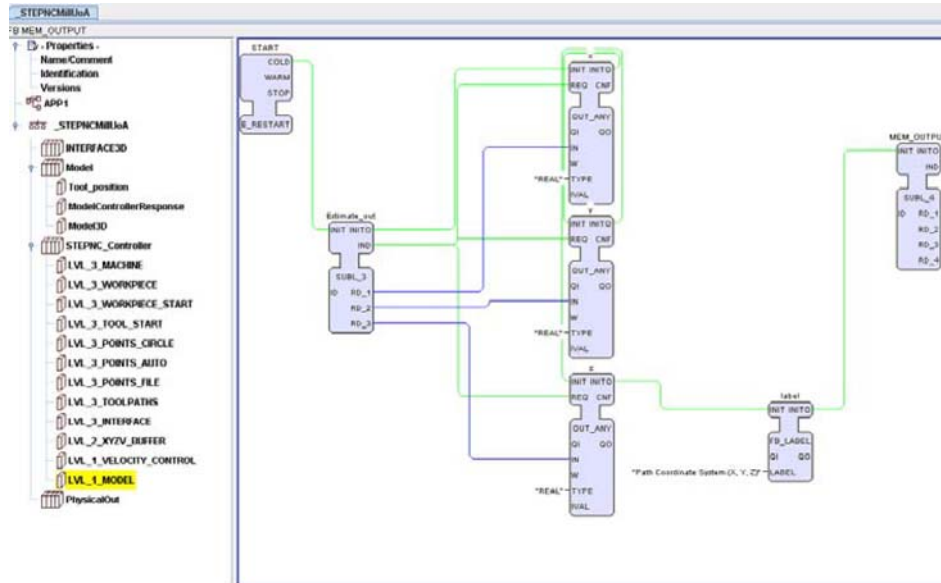
### 4.2 Support of simulation

The architecture of the generated FB system is based on the object-oriented MVC design pattern. According to the MVC pattern the simulation software configuration is a composition of three types of modules: Model, View and Controller. The use of this pattern allows a seamless transition from the simulation software configuration to the real control of physical machinery.

Figure 6 shows the system configuration that consists of four devices (in IEC 61499 terminology): Model, INTERFACE3D, STEPNC\_Controller and PhysicalOut, the first three corresponding to Model, View and Control as in the MVC pattern.



**Figure 6** Multilayered Function Block architecture of STEPNCMillUoA as seen in the Function Block editor of FBDK (see online version for colours)



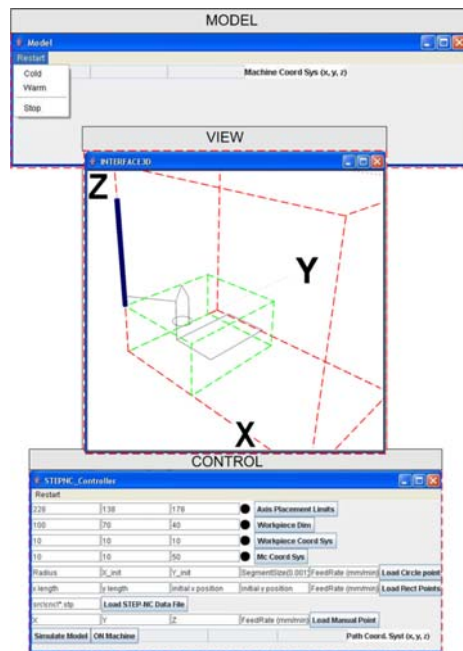
The elements of the architecture are explained as follows:

- The modelling modules in our case model the behaviour of the milling machine. Machine modelling is done by taking into account machine's structure, its physical dimensions, number of axes available and the type of inputs, outputs and movement expected. This module is allocated to a separate execution container (*Device* in the FB terminology). The MODEL device is further subdivided into three finer grain execution containers (resources): '*Tool\_position*', '*ModelControllerResponse*', and '*Model3D*' types, containing the FB applications, for modelling the Machine Coordinate System (MCS), workpiece dimension, tool travel limits, workpiece coordinate system, etc., based on STEP-NC data model.
- The task of the view module is to render status of the model or of the real machine. In the '*STEPNCMillUoA*' system configuration, the INTERFACE3D device contains one instance of the specialised '*ImageDev3D*' resource type to provide the necessary layout of view elements for the milling example. This device is also responsible for generation of the HMI.
- The Control module, implemented in the STEPNC\_CONTROLLER device contains the logic for sequencing and interlocking of operations among the modelled physical elements. Control is achieved through instances of the FB types Machine, Workpiece, Tool, Points, ToolPaths, Velocity\_Control and Model. With the usage of Physical output devices it outputs the signal to the motor drive axes to execute the machining instructions.

The output of the FB implementation of the CNC simulation is shown in Figure 7. At the moment, the prototype system only uses wireframe models for graphic representations. The cutting tool is represented by a cylinder. Lines represent the tool

path. The INTERFACE3D visualisation mirrors what one would expect of the milling machine. This allows for a preview of the controller motion, and can be used as a verification method.

**Figure 7** Output of the simulation (see online version for colours)



Source: Minhat et al. (2008)

The prototype system is implemented using a PC controlled CNC milling machine, a three-axis mill which was originally supplied with a PC-based Enhanced Machine Control (EMC) controller and a motor control unit that drives three stepper motors. The FB runtime software runs on the PC and enables the execution of STEPNCMillUoA. The very same code will run on embedded devices capable of supporting Java™.

### 4.3 System working procedure

Based on the design data of a part, the corresponding FB application is generated. The numeric data of the STEP-NC file are translated into parameters of the FBs. The geometry (contour) of a feature is represented as sequences of points which are then stored. Movement between the points is implemented by the motion controller FBs types, which will calculate an output to the axes motors from the sampled positions, generated by the calculated trajectory points. The output, generated from the motion controller is passed onto the discrete Input/Output (I/O) interface FBs, which send the control signals to the motor's drivers.

The process starts with STEP-NC data model of the part that is represented by the BTN\_ADD FBs type. It is then translated to the FB application Control (allocated in the 'STEPNC\_CONTROLLER' device). Execution containers (resources in IEC 61499

terminology) are used to separate these data out into distinct tasks consists of the machine (travel dimensions); Workpiece (dimensions); Workpiece\_Start; and Tool\_Start.

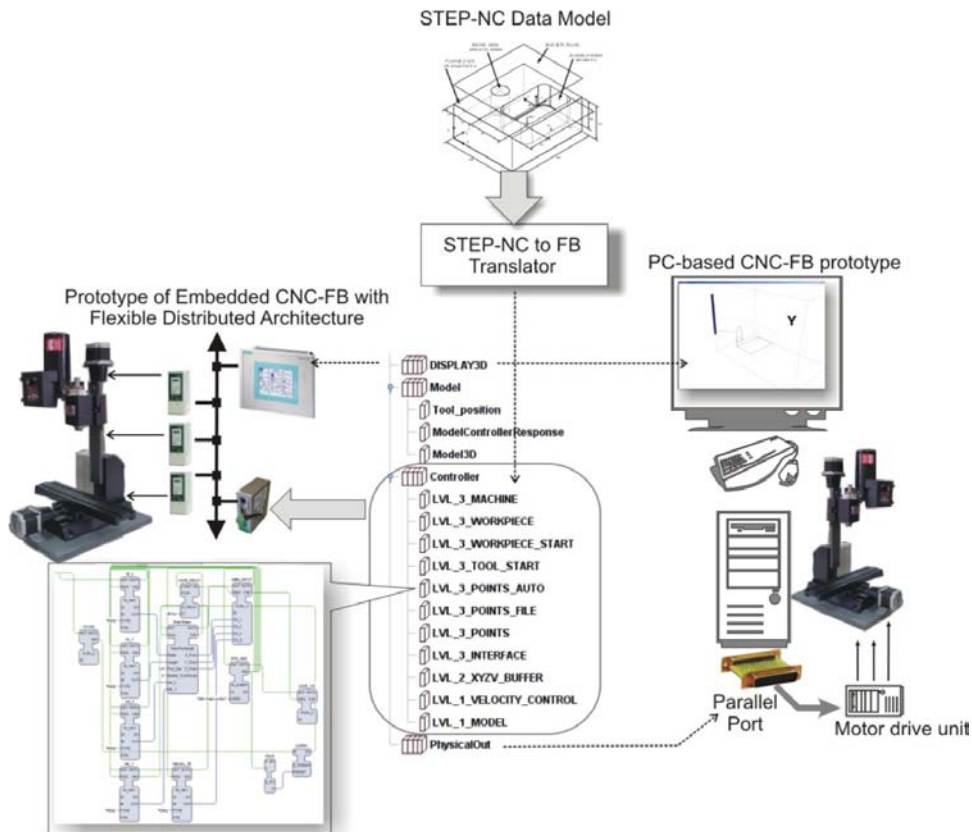
The 'Model' device contains the FB application, modelling uncontrolled behaviour of the machine tools. Given input control signals the model changes its state as the real machine tool does. It feeds the parameters of the current state to the device INTERFACE3D which can render FB applications.

The PhysicalOut device sends the control signals to the parallel port of a PC, on which the configuration is executed. The devices communicate with one another by means of communication FBs.

#### 4.4 Control of physical machine

The system application described with minimal changes can be executed on any computing platform supporting the IEC 61499 FBs from a personal computer with direct peripheral connection connected to a machine tool, to an embedded device with a distributed architecture, as illustrated in Figure 8. In the latter case, the motor drives, corresponding to the three axes and the display are connected to the main processing unit via a field area network (fieldbus).

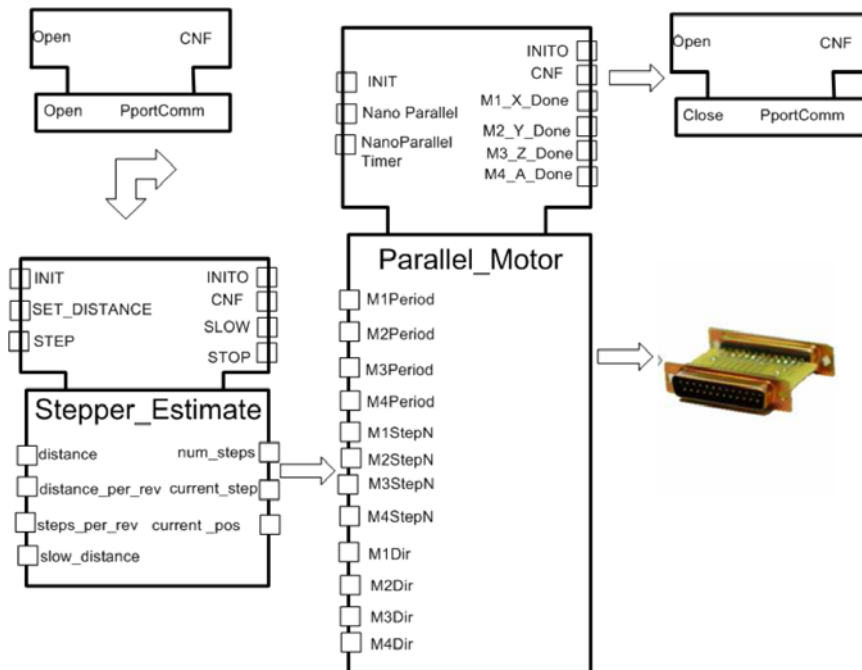
**Figure 8** Portability of Function Blocks enables deployment of STEP-NC – FB on various platforms (see online version for colours)



In our prototype system, the motor control board was used which was connected to the PC via a parallel port. In this configuration, only a direction signal and a single pulse are required to activate the motor in a given direction. The desired behaviour of physical interface is to utilise the raw data calculated in the controller (in terms of feed-rate) to control the real machine (i.e., to produce the signals required to drive the motors).

In this particular system, interfacing with hardware is dealt with by creating an abstract device model, called *ParallelPortDev*. The implementation of the device relies on the RXTX Java Library (2008), providing programming access to the hardware. The device model allows a specific library of FB to provide user interfaces to the motor drives via the parallel port of the PC. The access process includes opening the parallel port stream and calculating the required direction and number of steps to traverse. This is communicated to the *Parallel\_Motor* Function Block (Figure 9), which generates the required output sequence of data to execute the motor control, and in turn drives the motors on the machine. At the completion of the sequence the parallel port stream is closed.

**Figure 9** Parallel\_Motor Function Block (see online version for colours)



Source: Minhat et al. (2008)

A composite FB, using the services of the *Parallel\_Motor* FB, is designed to handle velocities and number of steps of all three motors. It can be easily extended to accommodate a fourth motor. This FB generates an output sequence which is written to the parallel port, and in turn drives the motors on the machine.

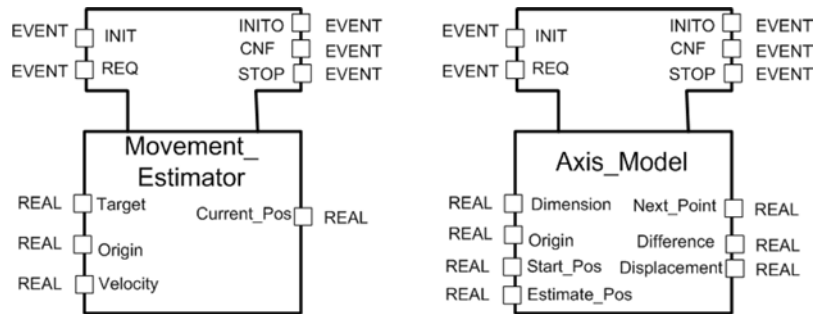
Geometrical parameters of the features are used as input parameters in the corresponding FB. The Workpiece data model will provide point-to-point connection to simulate the Tool Path Generator. The controller needs to generate two types of signals:

target signals and velocity signals. Target signals correspond to the coordinates (X, Y, and Z) of a destination, and the velocity signals represent the feed rate of each axis.

The *Movement\_Estimator* FB determines the data for the planned path, which is then fed into an axis model block that estimates the next point to be represented on the axis. The *Axis\_Model* block is used to represent a linear axis control unit to define an axis in the system by capturing its details. Therefore, in order to create the 3D representation of a model, three *Axis\_Model* blocks are needed to denote the (X, Y, and Z) coordinates of the next point. This block will take the speed and direction inputs for an axis and update its coordinate position. It can then be used for each of the X, Y and Z axis control (Figure 10).

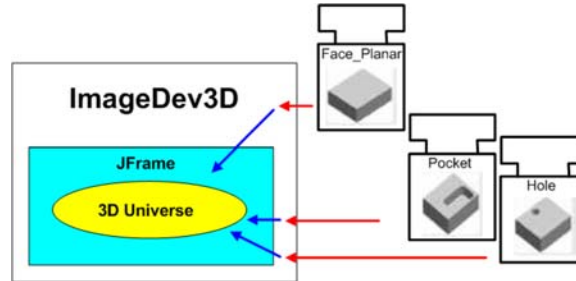
Further limits were added to the model to reflect the constraints of the machining boundaries. These include the maximum travelling speed and the maximum travel distances along each of the three axes.

**Figure 10** Movement\_Estimator and Axis\_Model Function Block



#### 4.5 Visualisation

The visualisation component of the MVC-based architecture of the STEPNCMillUoA is implemented in a 3D virtual environment through the Java™ extension, Java3D. The developed visualisation engine can render workpieces and cutting tools. Travel constraints and axes relationships are also modelled. The engine is accessible through the ImageDev3D device, and utilises a library of rendering FB types such as *RenderAxis3D*, *RenderWireBox3D*, *RenderCylinder3D* and *RenderLine3D*. 3D rendering is achieved in a three-step process. First a JFrame object is extended to support 3D rendering. This is done by creating a 3D Universe within it. JFrame itself is a native object in the 2D library Java.Swing, and handles the creation of a window object in which to render. Next, an interface to the FB environment is created. This interface is called the ImageDev3D. This is encapsulated as a FB device (as it represents a separate subsystem). An extended JFrame object is added as a child of the ImageDev3D. In the final step, separate FB are created that define a particular shape. These FBs construct Shape3D objects which can be rendered in the 3D Universe. The geometric data are transferred to the JFrame that exists in the ImageDev3D. The entire process is depicted in Figure 11. In this way it is possible to render multiple instances of any shape, as long as there is a FB for each instance. Furthermore it allows other more complex geometries to be created with ease.

**Figure 11** Visualisation by ImageDev3D rendering process (see online version for colours)

Each shape added to the 3D Universe is coupled with a transform object that allows a user to manipulate the translation, rotation and scale of the shape. Modifying the transform object allows dynamic motion of a shape without comprising the original shape data. In addition the 3D visualisation supports some typical navigation abilities as seen in many other 3D packages – namely rotate, zoom and pan. This enhances the user-friendliness of the interface.

#### 4.6 *Sample part*

The starting point of the process is STEP-NC data, which contains information such as Workplan, Workingstep, machining strategy, machining features and cutting tools. The prototype system was tested by manufacturing several parts, one of which is the first example in the annex of ISO 14649-11 in STEP Part 21 physical file.

Testing of the system was first completed using the INTERFACE3D visualisation. Following this the CNC milling machine was engaged with the same part information as used in the simulation. The resulting part matched the simulation in terms of dimensions of the shape machined as well as machining speed as shown in Figure 12.

**Figure 12** Sampled machined part and CNC milling machine (see online version for colours)

## 5 **Conclusions**

G-code only contains a small subset of the information that is generated by any CAD/CAM system tool. Its linear and circular movement instructions in a rigid sequential manner are not sufficient enough to support more flexible and

complex machine functionalities and modern graphical visualisation. The newly developed STEP-NC controller proved that when STEP-NC and FB are combined, it can yield benefits such as better user interface and improved functionality and operation mechanism. This G-code free CNC system runs on high-level data such as manufacturing features is capable of bi-directional communication and data flow in real-time, and allows easy modifications.

The CNC framework is developed using the concept of MVC, thus enabling full simulation and rendering of the CNC system, driven by the actual controller code. It integrates simulation of the model and visualisation of the controlled machine into the control system design process. The layered FB architecture provides for flexibility in adopting the controller to any new machines and computer hardware/software platforms. This is because that the layers are selected to separate the functional units into a hierarchical structure so that the controller becomes more readily maintainable, as modifications can be done to a particular layer without compromising the functionality of other layer. This research showcased a methodology to facilitate the transition from a centralised system to a distributed system and from a simulated system to a physical system. The STEP-NC controller making use of the IEC 61499-based MVC framework takes maximum advantage of its highly visual nature.

## Acknowledgement

The authors wish to acknowledge the contribution from S. Wong and Z. Al-Bayaa who developed the FB implementation of the STEPNCMillUoA.

## References

- Brecher, C., Vittr, M. and Wolf, J. (2006) 'Closed-loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks', *International Journal of Computer Integrated Manufacturing*, Vol. 19, No. 6, pp.570–580.
- Chouinard, J., Lavallee, D., Laliberte, J.F., Landreaud, N., Thramboulidis, K., Bettez-Poirtier, P., Desy, F., Darveau, F., Gendron, N. and Trang, C.D. (2007) 'An IEC 61499 configuration with 70 controllers; challenges, benefits and a discussion on technical decisions', *14th IEEE Conference on Emerging Technologies and Factory Automation (ETFA '07)*, Patras.
- Christensen, J. (2000) 'Design patterns for systems engineering with IEC 61499', in Doeschner, C.H. (Ed.): *Fachtagung Verteilte Automatisierung – Modelle und Methoded fuer Entwurf, Verifikation, Engineering und Instrumentierung*, Otto-von-Guenricke-Universitaet, Magdeburg.
- FBDK (2007) *Function Block Development Kit*, Online: <http://www.holobloc.com>, accessed February.
- Fortin, E., Chatelain, J.F. and Rivest, L. (2004) 'An innovative software architecture to improve information flow from CAM to CNC', *Computers and Industrial Engineering*, Vol. 46, No. 4, Spec. Iss., pp.655–667.
- Function Block Run-Time (FORTE) (2008) *4DLAC Consortium*, <http://www.fordiac.org/5.0.html>
- Grabmair, G., Zoitl, A., Strasser, T. and Froschauer, R. (2007) 'Modelling real-time constraints regarding reconfiguration aspects for IEC 61499 control applications', *IEEE International Conference on Industrial Informatics (INDIN '07)*, Vienna, Vol. 2, pp.1135–1140.

- Han, S., Liu, Z., Yang, X. and Zhao, J. (2007) 'Research on architecture of open CNC system based on component technology', *Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery*, Vol. 38, pp.127–131.
- Heusinger, S., Rosso Jr., R.S.U., Klemm, P., Newman, S.T. and Rahimifard, S. (2006) 'Integrating the CAx process chain for STEP-compliant NC manufacturing of asymmetric parts', *International Journal of Computer Integrated Manufacturing*, Vol. 19, No. 6, pp.533–545.
- Ho, K.H., Newman, S.T. and Allen, R.D. (2005) 'STEP-NC compliant information modelling for wire electrical discharge machining component manufacture', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 219, No. 10, pp.777–784.
- IEC 61499 (2005) *Function Blocks for Industrial-Process Measurement and Control Systems: Part 1: Architecture*.
- ISO 10303-21 (1994) *Product Data Representation and Exchange – Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure*.
- ISO 10303-238 (2007) *Product Data Representation and Exchange: Part 238: Application Protocol: Application Interpreted Model for Computerized Numerical Controllers*.
- ISO 14649-1 (2003) *Data Model for Computerized Numerical Controllers: Part 1 – Overview and Fundamental Principles*.
- ISO 14649-10 (2004) *Data Model for Computerized Numerical Controllers: Part 10: General Process Data*.
- ISO 14649-11 (2003) *Data Model for Computerized Numerical Controllers: Part 11 – Process Data for Milling*.
- ISO 14649-111 (2003) *Data Model for Computerized Numerical Controllers: Part 111: Tools for Milling Machines*.
- 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*.
- Kramer, T.R., Proctor, F.M., Xu, X. and Michaloski, J.L. (2006) 'Run-time interpretation of STEP-NC: implementation and performance', *International Journal of Computer Integrated Manufacturing*, Vol. 19, No. 6, pp.495–507.
- Lastra, L.M.J., Godinho, L., Lobov, A. and Tuokko, R. (2005) 'An IEC 61499 application generator for scan-based industrial controllers', *Proc. Third IEEE Conference on Industrial Informatics*, Perth, Australia, August, pp.80–85.
- Lee, W., Bang, Y.B., Ryou, M.S., Kwon, W.H. and Jee, H.S. (2006) 'Development of a PC-based milling machine operated by STEP-NC in XML format', *International Journal of Computer Integrated Manufacturing*, Vol. 19, No. 6, pp.593–602.
- Liu, R.L. and Zhang, C.R. (2004) 'Solutions for interpreting STEP-NC based part program', *Materials Science Forum*, pp.344–347.
- Mathews, C. (2002) 'Unlocking step-NC's potential', *Machine Design*, Vol. 74, No. 20, p.119.
- Minhat, M., Vyatkin, V., Xu, X., Wong, S. and Al-Bayaa, Z. (2008) 'A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks', *Robotics Computer Integrated Manufacturing*, Published online (doi: 10.1016).
- Nassehi, A., Newman, S.T. and Allen, R.D. (2006) 'The application of multi-agent systems for STEP-NC computer aided process planning of prismatic components', *International Journal of Machine Tools and Manufacture*, Vol. 46, No. 5, pp.559–574.
- Peng, F., Luo, Z., Li, B. and Chen, J. (2005) 'Interpretation of STEP-NC-based program', *Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition)*, Vol. 33, No. 8, pp.75–77.
- POSTECH Foundation (2008) Online, <http://e-mfg.postech.ac.kr>, access date, February.
- RXTX Java Library (2008) Online: <http://users.frii.com/jarvi/rxtx/index.html>, accessed January.
- STEP Tools, Inc. (2008) Online: <http://www.steptools.com/products/stdev/>, accessed January.



- Storr, A., Pritschow, G., Heusinger, S. and Azotov, A. (2002) 'Work step planning for turning with STEP-NC', *Arbeitsschrittplanung beim drehen mit STEP-NC*, Vol. 97, pp.390–396.
- Venkatesh, S., Odendahi, D., Xu, X., Michaloski, J., Proctor, F. and Kramer, T. (2005) 'Validating portability of STEP-NC tool center programming', *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference – DETC2005*, pp.285–290.
- Vyatkin, V. (2007) *IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design*, O3NEIDA – Instrumentation Society of America.
- Vyatkin, V. and Chouinard, J. (2008) 'On comparisons the ISaGRAF implementation of IEC 61499 with FBDK and other implementations', *Sixth IEEE Conference on Industrial Informatics*, Seoul, July.
- Wang, H., Xu, X. and Tedford, J.D. (2007) 'An adaptable CNC system based on STEP-NC and function blocks', *International Journal of Production Research*, Vol. 45, pp.3809–3829.
- Wang, L. (2002) 'DPP: a distributed process planning approach using function blocks', *Proceedings of the ASME Design Engineering Technical Conference*, Montreal, Quebec, Canada, pp.387–394.
- Wang, L., Song, C. and Feng, H.Y. (2006) 'Encapsulating detailed machining data in function blocks for adaptive manufacturing', *Transactions of the North American Manufacturing Research Institute of SME*, Vol. 34, pp.373–380.
- Xu, X.W. (2006) 'Realisation of STEP-NC enabled machining', *Robotics and Computer-Integrated Manufacturing*, Vol. 22, No. 2, pp.144–153.
- Xu, X.W., Wang, H., Mao, J., Newman, S.T., Kramer, T.R., Proctor, F.M. and Michaloski, J.L. (2005) 'STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration', *International Journal of Production Research*, Vol. 43, pp.3703–3743.
- Xu, X.W., Wang, L.H. and Rong, Y.M. (2006) 'STEP-NC and function blocks for interoperable manufacturing', *IEEE Transactions on Automation Science and Engineering*, Vol. 3, pp.297–308.
- Zhang, C., Liu, R. and Hu, T. (2006) 'On the futuristic machine control in a STEP-compliant manufacturing scenario', *International Journal of Computer Integrated Manufacturing*, Vol. 19, pp.508–515.
- Zoitl, A., Strasser, T., Hall, K., Staron, R., Surender, C. and Favre-Bulle, B. (2007) 'The past, present, and future of IEC 61499', *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4659 LNAI, pp.1–14.