Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 20 (2014) 103 – 108

www.elsevier.com/locate/procedia

2nd International Conference on Ramp-Up Management 2014 (ICRM)

A STEP-compliant Method for Manufacturing Knowledge Capture

X. Zhang*, B. Afsharizand, W. Essink, S. T. Newman, A. Nassehi

Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, United Kingdom

* Corresponding author. Tel.: +44(0)1225 384049; fax: +44(0) 1225 386928. *E-mail address:* x.zhang2@bath.ac.uk

Abstract

Over the last 50 years the development of CNC machines has seen a plethora of part programming languages. These programming languages have provided major barriers for the interoperability of information between CNC machines and Computer-aided (CAx) systems. Thus the process knowledge in existing part programs cannot easily be recycled and reused, due to an inability to interpret these forms of data. In fact, the process knowledge in the existing part programs is vital to develop process plans for new products, reduce leading time, and accumulate knowledge to enhance the product quality based on previous knowledge. In this paper, a STEP standard compliant method is proposed to recycle manufacturing knowledge from shopfloor in a universal manner. An EXPRESS model of CNC programming languages has been developed, though which different programing dialects can be translated into neutral data model. Based on the neutral data, a process comprehension method was used to capture process knowledge from CNC part programs and represent the knowledge in a standardised format. The EXPRESS model also forms a basis for a comprehensive machine tool modelling to assistant process planning activities. The proposed method is implemented in a prototype system and an industrial inspired component is used to validate the knowledge capture method.

Selection and peer-review under responsibility of the International Editorial Committee of the “2nd International Conference on Ramp-Up Management” in the person of the Conference Chair Prof. Dr. Robert Schmitt
Keywords: CNC; part program; resource modelling;knowledge management; STEP-NC

1. Introduction

With the rapid development of different machines and computer technology, CNC technology has advanced dramatically supporting those newly emerged machine functions. Compared with the first generation of Numerical Controlled (NC) machine, the modern CNC machines have got many advanced capabilities of such as multi-axis control, adaptive control, error compensation and multi-process manufacture (Nguyen and Stark 2005). With the versatility of CNC machines, the programming task becomes increasingly more difficult. For some precision and complex jobs, it is impractical to program at the shopfloor, which makes offline computer aided software tools a necessity for efficient generation and verification of CNC codes. These software packages are usually called Computer Aided Manufacturing (CAM) systems. CAM systems together with Computer Aided Design (CAD) systems and Computer Aided Process Planning (CAPP) systems make up a Computer Aided system (CAx) chain. Along the chain, an information flow is formed from CAx systems to CNC machines.

However, due to commercial interest, CNC controller vendors are very protective of every advance they made and employ proprietary standards for the enhancements that they introduced in their new controllers (Nassehi 2007). For example, CNC controller manufacturers introduced non-standard G-codes into the ISO 6983 standard to support their new features (additional axes, special caned cycles) resulting in various dialects for different machine and controller combinations (Proctor et al. 2002). Another reason why different dialects or languages are used to control the machine tool, the authors believe, is the simultaneous development of CNC machines tools around the world. People use different commands to program the same function. The standard (ISO6983-1 1982) of programming CNC machines actually came many years later after the first NC machine was launched and the standard is proposed based industrial practices.

Since CNC part programs are simple motion commands and switch operations, there is an information volume loss

Selection and peer-review under responsibility of the International Editorial Committee of the “2nd International Conference on Ramp-Up Management” in the person of the Conference Chair Prof. Dr. Robert Schmitt
doi:10.1016/j.procir.2014.05.038

during the post processors, as shown in Fig 1. At the shopfloor, the part programs are resource dependent, which means there is no interoperability between different CNC machines. The information loss, together with the fact that programming dialects are widely used, has prevented the information feedback from CNC part programs to CAD/CAM systems, especially in a uniform and automatic way (Newman *et al.* 2008). Since shopfloor is the final stage of the manufacturing activity, the part programs are valuable and the last, most accurate process plan used to machine the products. This onsite knowledge buried in thousands of part programs are difficulty to recycle and reuse.

Researchers (Liu et al. 2007; Schroeder and Hoffmann 2006) have tried to translate part programs and reuse them on different machines. The disadvantage of these solutions is that they only focus on interoperability between CNC machines, not between the CNC machine and the CAx chain. Direct translation of dialectic part programs is not an effective way since there are too many different part programming languages. It works only under the assumption that the two different machines involved in exchanging part programs have the same physical axis configuration and use cutting tools with the same diameters and cutting heights. Going beyond any of these assumptions would require the toolpath and the part program to be generated again in the CAD/CAM system (Zhang *et al.* 2013).

In this paper, a standard compliant method has been proposed to capture the shopfloor knowledge and make it available to be reused for new product development. The next two sections will present the process comprehension method for knowledge capture and CNC programming languages and machine tool modelling using a STEP-complaint method.

2. Machine capability profile with controller modelling

The manufacturing activities are currently relies on information technology in terms of CAx system. However, there is limited information about the manufacturing resources. In the state of the art, the nominal model of manufacturing resources is used, which fails to reflect the actual state of the resources(Newman and Nassehi 2009). Researchers have started to address the problem and proposed different approaches to model the manufacturing resources (Newman and Nassehi 2009; Zhang *et al.* 1999; Vichare *et al.* 2009). One disadvantage of these modelling approaches is the fact that the resource models focus on the hardware (machine tools, cutting tools, auxiliary devices etc.) without considering the software such as controller system, programming languages. The software information is an important part of the resource model and plays an important role in utilising the resource model for decision making during the manufacturing process.

In this research, an EXPRESS model of CNC programming languages has been developed, which covers most commands used on 3-axis milling machines. The programing languages model serves as an important part of CNC controller modelling, whereas the controller model is part of the machine capability profile (Afsharizand *et al.* 2014). Using this model, different programming languages/dialects can be represented by a neutral data format for implementation of a standardised knowledge capture method in the next stage. Otherwise, there is a need to develop separate interpreting interfaces for each type of CNC part programming dialects for different controllers. This work should be huge and not practical for implementation since there are thousands of programming dialects in (Maeder *et al.* 2002; STEP Tools Inc 2012). Also, the modularisation, to have separate modules of inputting and handling data, helps to make the system less complex, loose coupling between modules and easy to implement and test.

The model has been developed in EXPRESS language, which has been used in series of STEP standards. The modelling method using EXPRESS will keep the process comprehension system consistent with existing STEP stands and will be helpful for wide implementation in the later stages. Based on the EXPRESS schema, each controller's programming specification (syntax) can be described in STEP-21 files. Part programming in different programming dialects can be translated into a Meta-data, as shown in Figure 2, using a single translation interface. The STEP-21 specification files can then form a dictionary of controllers (programming specifications). The dictionary mechanism will eliminate the need to develop plenty of translators for different programming dialects.

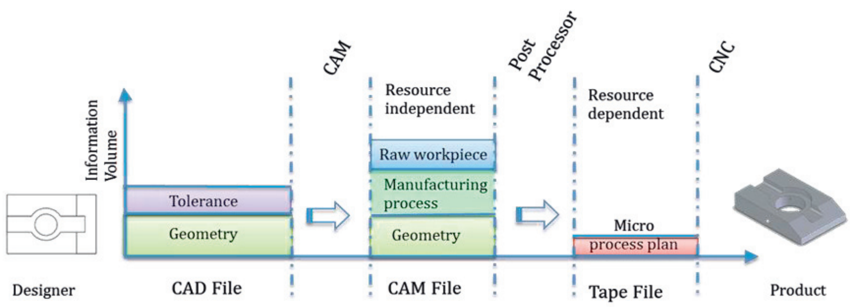

Figure 1 - Information lost during the process lifecycle

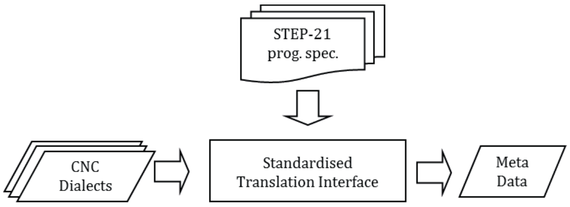

Figure 2 CNC dialects translation to Meta-data

The EXPRESS-G diagram of the EXPRESS model of CNC programming languages is shown in Figure 3. In the model, NC_programe_specification entity is the root entity for each dialect. The programming syntax will be specified for each Instruction in its_grammar. For each Instruction, there can be a set of the programming syntax for the cases that some commands have different programming formats. For example, for circular interpolation, there are usually two methods to define the arc: by centre point or by radius. The Instruction has been classified into two groups: Interpolation and NC_function. Each Instruction has an attribute of modal, which can be set to true of false to indicate the commands will be continually in effect or not without implicit programmed for each line. In this diagram, only part of the CNC instruction has been included.

Based on the EXPRESS model of CNC programming languages, a programing specification of a Fanuc controller has developed, as shown in Figure 4. From the figure, the specification is developed for Fanuc 18i controller. The command's SYNTAX presents the programming format and the PARAMETERS of each syntax defines the parametric elements for the command syntax. For Example, the parameter 'R@RADIUS' of CW_CIRCULAR is used to define the radius of the arc of a clockwise circular interpolation. There are two types of parameters: optional and Mandatory, to indicate whether the parameter can be omitted. In this specification, there are two SYNTAX entities for circular interpolation, as mention before in this section, to define two programming formats of circular movement.

As mentioned, the model of the programming language serves as a part of the machine capability profile, which aims to provide an overall modelling of a machine tool. The profile can capture the dynamic status of the machine tool and can be used to access the current machining capability of the machine to support the decision making in the manufacturing activities (Afsharizand *et al.* 2014). As shown in Figure 5, an excerpt the capability profile of a Dugard machine has been illustrated. In the capability profile, cutting tool and CNC controller have been included. The programming specification of the controller (Fanuc 18i), which is in a separate file in the same directory, is linked into the capability profile (#28).

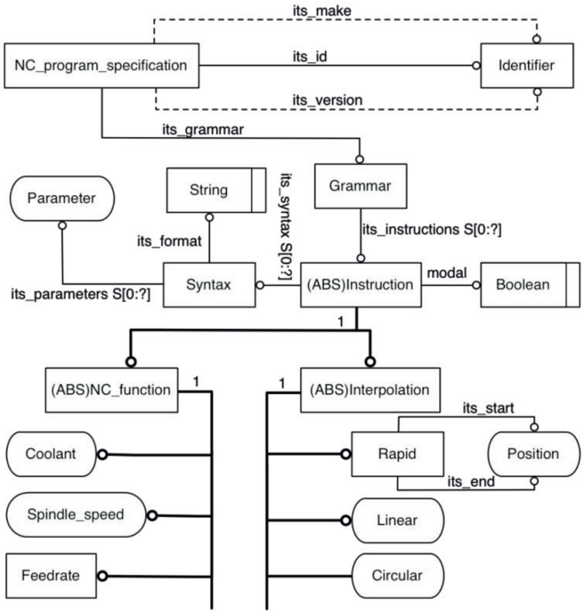

Figure 3 EXPRESS-G diagram of programming specification

Fanuc_schema.txt

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('NC PROGRAMMING SCHEMA','FANUC CONTROLLER'), '1');
FILE_NAME('FANUC.STP', '2013-02-25', ('XIANZHI ZHANG'),('UNIVERSITY OF BATH, BATH,UK'),$,$,$);
FILE_SCHEMA(('NC_PROGRAMMING_SCHEMA'));
ENDSEC;

DATA;
#1=NC_PROGRAME_SPECIFICATION('FANUC18I','BETA1','FANUC',#2);
#2=GRAMMAR((#4,#10,#15,#20,#25,#28,#31),(#60,#90,#95,#97,#100));
#4=RAPID(.T.,(#5));
#5=SYNTAX('$*G0*$X@AXIS0$Y@AXIS1$Z@AXIS2',#6);
#6=PARAMETERS(('$G0'),());
#10=LINEAR(.T.,(#11));
#11=SYNTAX('$*G1*$X@AXIS0$Y@AXIS1$Z@AXIS2',#12);
#12=PARAMETERS(('$G1','$F@FEEDRATE'),());
#15=CW_CIRCULAR(.T.,(#16,#18));
#16=SYNTAX('$*G2*$X@AXIS0$Y@AXIS1$Z@AXIS2$R@RADIUS$H@HELICAL$F@FEEDRATE',#17);
#17=PARAMETERS(('$G2','$F@FEEDRATE','R@RADIUS','H@HELICAL'),());
#18=SYNTAX('$*G2*$X@AXIS0$Y@AXIS1$Z@AXIS2$I@R1$J@R2$K@R3$H@HELICAL$F@FEEDRATE',#19);
#19=PARAMETERS(('$G2','$F@FEEDRATE','I@CX','J@CY','K@CZ','H@HELICAL'),());
#20=CCW_CIRCULAR(.T.,(#21,#23));
```

Figure 4 Fanuc programming specification based on CNC language model

```
DATA;
#1=MACHINING_CAPABILITY_PROFILE(#22,(#2));
#2=MACHINING_CAPABILITY_DATA_POINT('"DUGARD CAPABILITY PROFILE"',(#5,#6,#7,#8,#9),#3,#4);
...
#7=TOOLING_CAPABILITY(13.0,66.0,30.0,(#13,#16,#19));
...
#13=CUTTING_TOOL_INFO(1,$,$,(),'"ENDMILL"',#14,150.0);
#14=MILLING_TOOL_BODY(#15,2,$,$,$);
#15=MILLING_TOOL_DIMENSION(10.0,$,$,$,$,$,$);
#16=CUTTING_TOOL_INFO(4,$,$,(),'"FACEMILL"',#17,170.0);
#17=MILLING_TOOL_BODY(#18,2,$,$,$);
#18=MILLING_TOOL_DIMENSION(40.0,$,$,$,$,$,$);
...
#22=MACHINE_TOOL_SPECIFICATION('"DUGARD"',.MILLING_MACHINE.,#26,(#27),$,$,#23,#28,$,(#29,#30,#31,#32));
...
#28=NC_CONTROLLER('"18i"','"FANUC"',.INCH_AND_METRIC.,5.0,4.0,1.0,0.0010,0.01,2.0,'FANUC_SCHEMA.STP',(),$,
...
```

Figure 5 Excerpt of a machine capability profile

3. Process comprehension method for knowledge capture

To capture process knowledge from G&M code using the machine tool profile, a process comprehension system (Zhang *et al.* 2013) has been developed. The input of the system includes G&M codes and the geometry of workpiece used to machine the part. Referring to the machine capability profile the process comprehension system can retrieve the controller information which includes cutting tool data (tool diameter) and programming specification. Part programs can be translated into a set of neutral data with the help of the controller programming specification. Based on the neutral data including tool path, cutting parameters, machine functions etc. translated from part programs, a feature recognition method has been developed to recognise 2½D milling features. The feature recognition algorithm calculates the tooth path considering the tool size to compute the feature geometries. Based on the machining parameters and machining strategies, the corresponding machine operations of the features can be identified as well. By organising the features and operations into corresponding STEP-NC entities, a STEP-NC part21 file will be generated to represent the process knowledge in the original part programs.

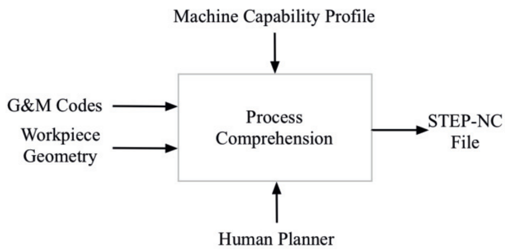

Figure 6 IDEF-0 diagram of process comprehension

4. Case study

An industrial inspired test component as shown in Figure 7 has been used to validate the manufacturing knowledge capture method proposed in this paper. In this component, different features including planar face, boss, pocket and hole have been covered. The part was designed and processed in commercial CAD/CAM system and the part programme for Fanuc controller has been generated. Using the part programme as the input for process comprehension system, the features and associated operations has been recognised. As shown in Figure 8, there are 14 features has been recognised. Based on that, a STEP-21 file representing the original process plan has been generated as shown in Figure 9. As shown, there are 10 workingsteps/operations (2) to machine the component. The workpiece is a fillet with the size of 135 x 185 x 40 mm (1). The depth for the first face milling operation is -2mm (4). The cutting parameters are included in the milling technology entity. For the first operation, the feed rate is 0.00417 m/s and the spindle speed is 33.15 rps (Revolutions Per Second).

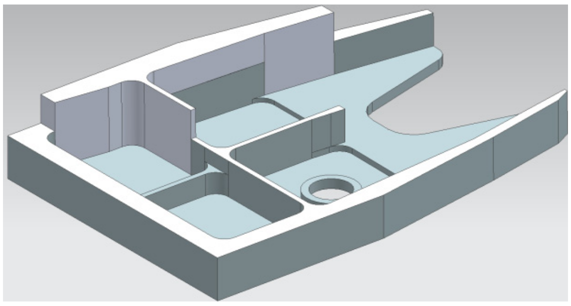

Figure 7 Test component for validation

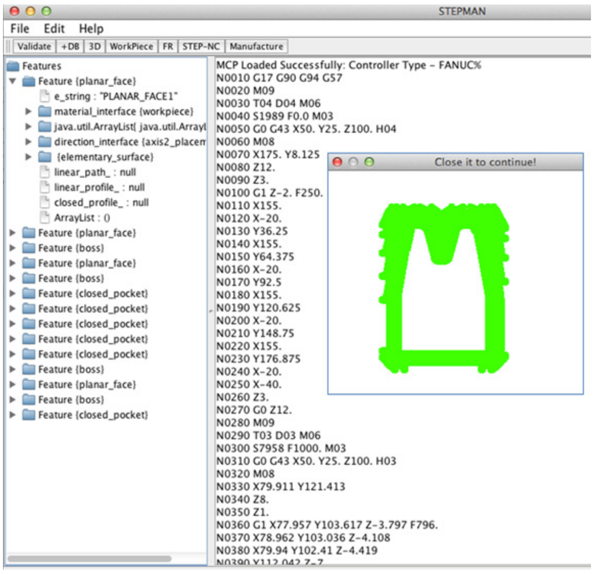

Figure 8 Process comprehension of test component

5. Conclusions and future work

Due to the existence of many different part programming languages and the mechanism of part program generation, the shopfloor knowledge cannot be captured automatically to reuse on new product development. In this paper, a STEP-compliant approach has been proposed to model the "soft" manufacturing sources, which are the programming languages of the CNC controllers. EXPRESS modelling language has been used to ensure the conformance with the other "hard" resource modeling (machine capability profile). The machine capability profile incorporating the controller languages model will provide a comprehensive description of CNC machines. With the model of CNC programming languages, the part programs written in different languages can be represented by a neutral data, which will be processed in a generic way to extract the process knowledge from shopfloor.

A simplified fish head part has been used to validate the model of CNC languages and the knowledge capture method. The test results show that the model is capable to represent the process knowledge in part programs and process comprehension can extract the original knowledge. The presentation of the process knowledge was organized in STEP-21 file, which is a standardised format and could be reused by other systems.

The future work will focus on the development of machine capability profile and the exploration of other usage of the profile to provide support for decision makings. One area will be utilising the capability profile to access the machining capability of a machine against a specific component machining, and to provide capability reports for companies by combining capability profiles of individual machine. The other area can be knowledge based process planning to take advantage of the process knowledge extracted from part programs.

Acknowledgements

This research has been supported by European Union's Seventh Framework Programme under grant agreement n286962 (the STEPMAN project). The authors gratefully express their thanks for the advice and support of all concerned.


```
ISO-10303-21
HEADER;
FILE_DESCRIPTION(('STEPMAN generated STEP file'),'1');
FILE_NAME('FISH_HEAD.STP',
('Behnood Afshari','Wesley Essink','Xianzhi Zhang','Aydin Nassehi'),
('University of Bath'),' ','STEPMAN',' ');
FILE_SCHEMA(('COMBINED_Schema'));
ENDSEC;
DATA;
#1=PROJECT('RECOGNISED ISO 14649 PART 21  FILE FROM G&M CODES',#2,(#3),$,$,$);
#2=WORKPLAN('MAIN WORKPLAN',(#4,#5,#6,#7,#8,#9,#10,#11,#12,#13),$,#14,$);
#3=WORKPIECE('WORKPIECE135.0X185.0X40.0',$,$,$,$,#24,());
#4=MACHINING_WORKINGSTEP('WS PLANAR_FACE1',#15,#16,#17,$);
#5=MACHINING_WORKINGSTEP('WS PLANAR_FACE2',#15,#41,#42,$);
#6=MACHINING_WORKINGSTEP('WS PLANAR_FACE3',#15,#120,#121,$);
#7=MACHINING_WORKINGSTEP('WS POCKET1',#15,#216,#217,$);
#8=MACHINING_WORKINGSTEP('WS POCKET2',#15,#289,#290,$);
#9=MACHINING_WORKINGSTEP('WS POCKET3',#15,#358,#359,$);
#10=MACHINING_WORKINGSTEP('WS POCKET4',#15,#431,#432,$);
#11=MACHINING_WORKINGSTEP('WS POCKET5',#15,#500,#501,$);
#12=MACHINING_WORKINGSTEP('WS POCKET6',#15,#597,#598,$);
#13=MACHINING_WORKINGSTEP('WS PLANAR_FACE4',#15,#618,#619,$);
#14=SETUP('SETUP',#792,#15,(#793));
#15=ELEMENTARY_SURFACE('SECURITY PLANE',#18);
#16=PLANAR_FACE('PLANAR_FACE1',#3,(#17),#22,#23,$,$,$,());
#17=PLANE_MILLING($,$,'PLANAR_FACE1',$,$,#29,#30,#31,$,$,$,$,$,$);
#18=AXIS2_PLACEMENT_3D('SECURITY PLANE PLACEMENT',#19,#20,#21);
#19=CARTESIAN_POINT('SECURITY PLANE: LOCATION',(0.0,0.0,10.0,0.0,0.0));
#20=DIRECTION('AXIS',(0.0,0.0,1.0));
#21=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#22=AXIS2_PLACEMENT_3D('PLANAR_FACE1 PLACEMENT',#34,#35,#36);
#23=ELEMENTARY_SURFACE('PLANAR_FACE1 DEPTH PLANE',#37);
#24=BLOCK('WORKPIECE BLOCK',#25,135.0,185.0,-40.0);
#25=AXIS2_PLACEMENT_3D('WORKPIECE BLOCK PLACEMENT',#26,#27,#28);
#26=CARTESIAN_POINT('WORKPIECE BLOCK: LOCATION',(0.0,0.0,0.0));
#27=DIRECTION('AXIS',(0.0,0.0,1.0));
#28=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#29=MILLING_CUTTING_TOOL('T4',#32,(),$,$,$);
#30=MILLING_TECHNOLOGY(0.004166666666666667,.TCP.,$,33.15,$,$,$,$,$);
#31=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#32=FACEMILL(#33,$,$,$,$);
#33=MILLING_TOOL_DIMENSION(40.0,$,$,$,0.0,$,$);
#34=CARTESIAN_POINT('PLANAR_FACE1',(175.0,8.125,0.0));
#35=DIRECTION('AXIS',(0.0,0.0,1.0));
#36=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#37=AXIS2_PLACEMENT_3D('PLANAR_FACE1 DEPTH',#38,#39,#40);
#38=CARTESIAN_POINT('PLANAR_FACE1 DEPTH',(0.0,0.0,-2.0));
#39=DIRECTION('AXIS',(0.0,0.0,1.0));
```

Figure 9 STEP-NC presented process plan

References

[1] Afsharizand, B., Zhang, X., Newman, S. and Nassehi, A., 2014. Determination of machinability considering degradation of accuracy over machine tool life cycle. Proceedings of the The 47th CIRP Conference on Manufacturing Systems, Windsor, Canada. 28-30 April.

[2] ISO6983-1, 1982. Numerical control of machines—program format and definition of address words, Part 1: data format for positioning, Line motion and contouring control systems. International Standards Organisation (ISO).

[3] Liu, Y., Guo, X., Li, W., Yamazaki, K., Kashihara, K. and Fujishima, M., 2007. An intelligent NC program processor for CNC system of machine tool. Robotics and Computer-Integrated Manufacturing, 23(2), pp.160-169.

[4] Maeder, W., Nguyen, V.K., Richard, J. and Stark, J., 2002. Standardisation of the Manufacturing Process: the IMS STEP-NC project. Proceedings of the INational Network of Competence on Integrated Production Logistics Workshop, Saas Fee, Switzerland. pp.5.5/1-5.5/3. Sep. 10-12.

[5] Nassehi, A., 2007. The realisation of CAD/CAM/CNC interoperability in prismatic part manufacturing. University of Bath, Bath, UK.

[6] Newman, S.T. and Nassehi, A., 2009. Machine tool capability profile for intelligent process planning. CIRP Annals - Manufacturing Technology, 58(1), pp.421-424.

[7] Newman, S.T., Nassehi, A., Xu, X.W., Rosso Jr, R.S.U., Wang, L., Yusof, Y., Ali, L., Liu, R., Zheng, L.Y., Kumar, S., Vichare, P. and Dhokia, V., 2008. Strategic advantages of interoperability for global manufacturing using CNC technology. Robotics and Computer-Integrated Manufacturing, 24(6), pp.699-708.

[8] Nguyen, V. and Stark, J., 2005. STEP-compliant CNC Systems, Present and Future Directions, Advanced Design and Manufacturing Based on STEP. Springer.

[9] Proctor, F.M., Michaloski, J.L. and Shackleford, W.P., 2002. Tying together design, process planning and machining with STEP-NC technology. Proceedings of the 5th Biannual World Automation Congress, Orlando, Fl. pp.33-38. Jun 09-13.

[10] Schroeder, T. and Hoffmann, M., 2006. Flexible automatic converting of NC programs. A cross-compiler for structured text. International Journal of Production Research, 44(13), pp.2671-2679.

[11] STEP Tools Inc,2012. STEP AP 238 FAQ: Questions about Manufacturing Data Standards [Online]. Available from [http://www.steptools.com/library/stepnc/faq/faq_02.html].

[12] Vichare, P., Nassehi, A., Kumar, S. and Newman, S.T., 2009. A Unified Manufacturing Resource Model for representing CNC machining systems. Robotics and Computer-Integrated Manufacturing, 25(6), pp.999-1007.

[13] Zhang, X., Nassehi, A., Safaieh, M. and Newman, S.T., 2013. Process comprehension for shopfloor manufacturing knowledge reuse. International Journal of Production Research, pp.1-15.

[14] Zhang, Y., Feng, S.C., Wang, X., Tian, W. and Wu, R., 1999. Object oriented manufacturing resource modelling for adaptive process planning. International Journal of Production Research, 37(18), pp.4179-4195.