

Conceiving of a High Level Language for Programming of the Machine Reconfiguration and Part Machining Dedicated to the New Reconfigurable Machine - Tools Generation

MARIN F.B., EPUREANU A.I., MARINESCU V., BANU M., CONSTANTIN I.
"Dunărea de Jos" University of Galați

ABSTRACT

This paper is proposing a new approach of the RMT (Reconfigurable Machine Tool) programming. The approach is based on a new control architecture characterized by high level of reconfigurability. The programming consists in the building of two documents: i) machine-program, concerning machine reconfiguration phase and ii) part-program, concerning the programming of the machine to process a specific part. A visual language architecture is proposed for the two documents, allowing a fast programming and a fine-grain control. This visual language is replacing the traditional G-code. In addition, the language allows the programming of the machine hardware reconfiguration, according to the current part family to be processed.

Keywords: Reconfigurable machine tool, CAM software, numerical control reconfiguration, CNC programming.

1. Introduction

Nowadays, companies building manufacturing machines must face fast changes happening on technical, commercial and economical fields. These emerging evolutions includes economy globalization, individualization of needs, capital dynamization, generating high requirements concerning investment efficiency and high versatility of the small companies. At the time, the companies responses to these changes are based on the idea of extending some of the architectural characteristics of the classic manufacturing system to become more reconfigurable. Whereas Dedicated Manufacturing Systems (DMSs) have been designed to produce a specific part and Flexible Manufacturing Systems (FMSs) are designed to accommodate a large variety of parts even though the parts are not specified at the system design stage, the Reconfigurable Machine Tools (RMTs) combine the advantages of DMSs and FMSs providing an suitable solution [1][2]. These machine tools are designed to be adapted for the mixed products and variable volume demands. The reconfigurable machine is conceived to be build using standard and inter-operable components, including "plug-and-play" hardware modules and universal fixtures and tools assembled into

arbitrary architecture with needed degrees of freedom.. The issues involved in the reconfiguration process are represented by:

- hardware modules reconfiguration;
- software reconfiguration (both programming and control).

The modeling algorithms of the reconfigurable machine kinematics need to be developed. These algorithms are the basis for the control and simulation of reconfigurable machine tool. By simulation of the hardware and control architecture, it is assured the optimal configuration for the new reconfigured machine.

The concepts of the control paradigms, such as holonic manufacturing, bionic manufacturing [2], fractal companies [1], are proposed for the next-generation manufacturing systems. So far that concern the control concept, it was proposed a software architecture based on a combination of object-oriented models and executable formal specifications [13]. As the authors present, in its proposed architecture the machine tool control software is composed of a set of reusable software components. Each component is modeled using a set of event-based external interfaces, a control logic driver for execution of behavioral specifications, and a set of service protocols for platform adaptation. The entire software

behaviour can be viewed as an integration of behaviours of components.

In response to such challenges, various efforts have been made, such as Open Modular Architecture Controllers (OMAC) [11], Open System Architecture for Controls within Automation Systems (OSACA) [12], Open System Environment for Controllers (OSEC), Hierarchical Open Architecture Multi-processor for CNC (HOAM-CNC), Open Architecture Controller (UMOAC) etc. The common approach in these researches is the adoption of an open, modular architecture to promote interoperation [13].

Traditionally, CAM has been considered as an NC programming tool used to generate CNC program (commonly G-code) to drive numerical controlled machine tools. Most of current numerical control systems adopt a architecture as described in Fig.1. A computer system, which is usually a general-purposed personal computer (PC) or another kind of a computer, is chosen as the NC core system platform. Human interface software includes functions such as NC code editing and decoding.

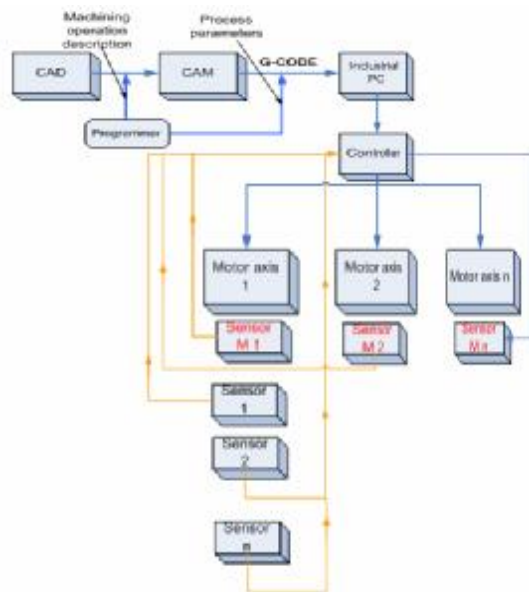


Fig.1 Present CNC control architecture

Functions such as interpolation and control, cutter radius compensation, general I/O or logic control, servo control, etc., are implemented in several controllers (such as PLC – Programmable Logic Controller). Consequently, the controllers are managing commands towards variators to control motors and to read data from sensors.

Hardware reconfiguration means the building of a new machine using the hardware

modules to process a new part-family. It is a must have to develop architectures and techniques concerning software reconfigurability to provide short ramp-up time.

The issue of reconfiguring the machine in a new architecture involves great challenges so far that concern the control and programming. The time needed for reconfiguring the control needs to be as limited as possible, otherwise the advantages of the RMT concept will vanish.

The nowadays CAD/CAM systems are inappropriate for the concept of RMT. Also, the actual common G-code language for CNC programming is not suitable for programming RMT.

Several disadvantages as long path addressing, ambiguous semantics in some cases, vendor-specific extensions, difficulty of the in-process changes, use of specific post-processors, are just a few to note.

In our previous works [15][16] we proposed a new control and hardware architecture to satisfy reconfigurability requirements. The objective of this paper is to develop a high level language to describe the hardware reconfiguration, and secondly to describe the machine workcycle, specific for a given part.

The language will be used to build the two technological documents namely i) machine-program, concerning machine reconfiguration phase and ii) part-program, concerning the programming of the machine to process a specific part.

The remainder of the paper is organized as it follows. In Section 2 it is compared the traditional and the new proposed control and hardware architecture. In Section 3 it is presented the proposed programming language

Finally section 4 gives conclusions of this investigation.

2. The new control and hardware architecture to which the proposed language is dedicated.

According to our approach a bus is handling communication for all modules [Fig. 3]. Also, compared to today's fieldbus architecture, the hardware modules have close-loops control represented by integrated controller able to communicate with industrial computer.

The information sent to hardware module is represented by the required position of the hardware module only. Moreover, if a hardware module has specific sensors besides position control of its movement, this is handled by the integrated controller on the hardware module.

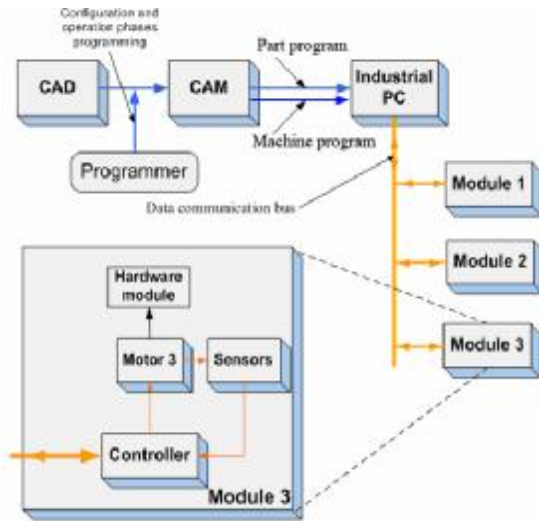


Fig. 2. The proposed control architecture

Hardware modules, as seen in Fig.3, are composed of efector, sensors and motor, which are controlled by the controller embedded on the module. The controller is controlling the hardware module only. Compared with nowadays CNC architecture, where the PLC controller is handling simultaneously motors control and different sensors in the system, the embedded controller is handling only the hardware module. This controller is communicating with the PC using a common bus for all hardware modules. The PC is running several independent operating system processes that are controlling the hardware

modules by communicating to embedded controller.

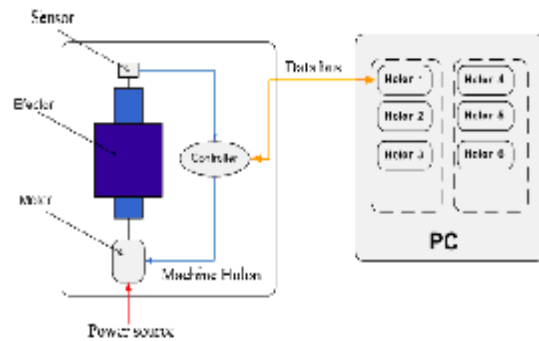


Fig. 3. Generic holon architecture

Compared to nowadays CAM architecture, the proposed adaptive-CAM (A-CAM) is not providing source code (G-code) to be interpreted and executed by machine, but an information package to describe the machine configuration and operation phases. Whereas present CAD/CAM software outputs source code describing geometric path, and none or few information concerning machining parameters (such as federate), our A-CAM architecture is providing the parametric algorithm of control variables online scheduling.

This algorithm is run in RSU (Reactive Scheduling Unit) [Fig.4]. Whereas in the process design phase the algorithm is run with nominal values of its parameters algorithm constants, on the machine is running with updated values of algorithm parameters.

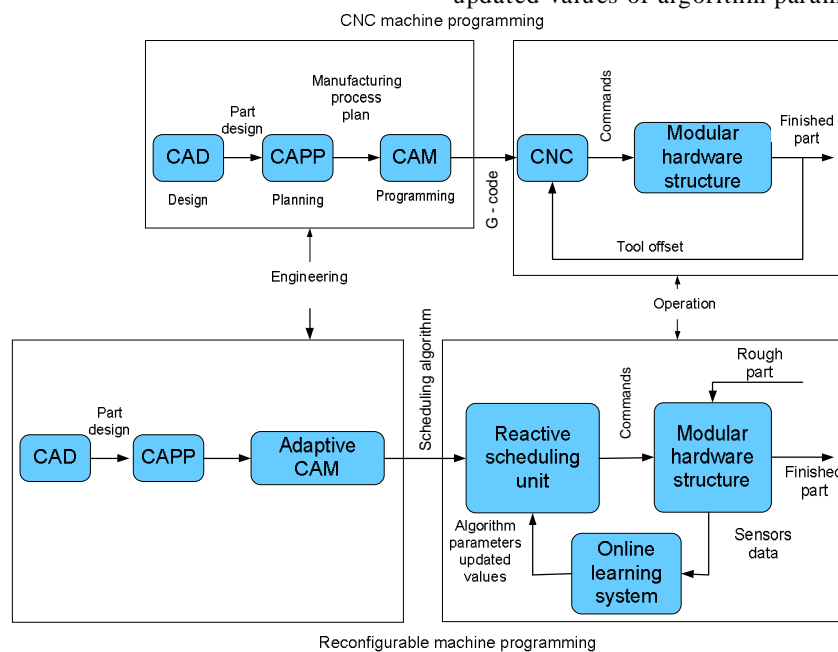


Fig. 4. The proposed versus present CAPP/CAD/CAM/CNC architecture

3. The proposed programming language

Machine programming according to the new view is concerning of two distinctive phases of machine existence. Corresponding with the building the new machine phase using the hardware modules, in order to process a part family; it is the machine-program.

Similarly, corresponding to part processing phase, it is the part-program. As described in Fig. 5 both programs are developed in a visual environment allowing easy programming.

Using solid model in a 3D virtual environment the programmer builds using different modules the kinematic model of the machine (machine-program) and then, after a virtual machine is build, the programmer describe the trajectories to process the part (part-program). In order to build the part-program the programmer describe each part of the trajectory with the some technical requirements (called task according to our approach) and virtually move the machine parts to execute each task. This way, it is tested collision issues and the part is virtually processed.

The hardware configuration unit (Fig. 5) is the software module to be used for machine-program developing, whereas software product running unit is the software module used to develop part-program.

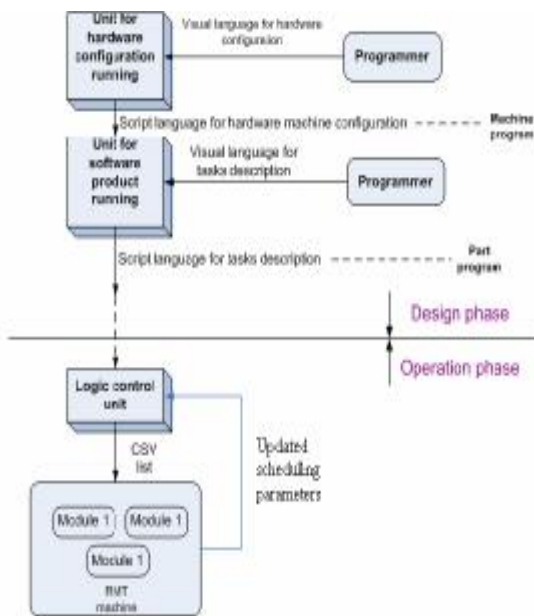


Fig. 5 Machine – program and part program

Machine program

Data processing algorithm to build machine-program is composed of several phases to be performed by programmer as described bellow:

- 1) Imports part CAD model to be processed.
- 2) Divide tool trajectory in several parts.

According to the proposed approach these parts are the tasks to be accomplished.

3) Imports different hardware CAD models and tries different configurations by simulating trajectories implies in part-processing.

4) Choose those modules that allow every task accomplishment.

Machine-program comprises of several information sets (Fig. 6) as it follows:

i) *script program*, which is generated automatically by visual programming environment used by programmer to simulate and chose machine hardware configuration. Also to the scripting program is included:

ii) *digital image of machine hardware configuration*, representing information concerning fixturing information for an easy understanding by operators assembling new machine.

iii) *module list* to be used and their characteristics.

Building the part-program implies knowing the geometric and kinematics characteristics of every module (maximum feed rate, maximum momentum of the motor). Bellow is given an example for the script image to be generated in the visual environment:

```
WorldReference(X,Y,Z)
Set.Module23.At(X,Y);
Set.Module45.On.Module23.At(X,Y);
Set.Module98.At(X,Y);
Set.Module45.On.Module98.At(X,Y).Angle(FX).CW
```

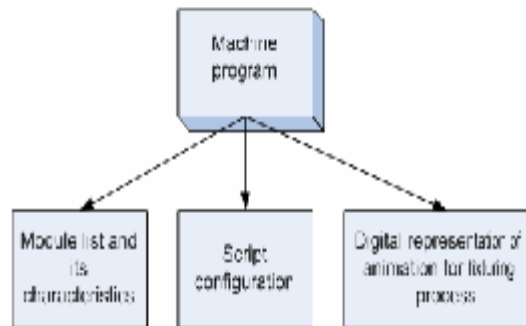


Fig. 6 The machine-program

Part-program

After building the machine-program it is necessary to build part-program which consists in the following steps (Fig.9):

- 1) Imports machine-program, describing the current machine configuration.
- 2) Imports CAD model of the blank and divide the trajectory in several parts, defining in this manner the tasks.
- 3) Inputs nominal CAD of finished part
- 4) Inputs for each task the requirements imposed to the finished part for every surface that is describing each task (requirements such as roughness and tolerance).
- 5) Inputs blank material characteristics.
- 6) Imports CAD model of the tool.
- 7) Tests and simulates process, considering blank profile nominal CAD model (during processing, the blank is measured on machine). This testing aims to identify successive tasks to avoid collision issue and to find suitable tools for the current processing.
- 8) It is build automatically the information package representing part-program.

The programmer is not inputting the machining parameters describing the process, but only the successive tasks and the requirements imposed to the finished part. The some, he doesn't describe the task in a scripting manner, but describes it in a visual environment, by placing in a 3D environment the tools in the machine current hardware configuration, in order to satisfy collision conditions. Consequently he fills in for every task defined the requirements imposed for each surface.

Technological document called part-program is composed of four digital information (Fig. 7): CAD file data, script operation image, control algorithm for each task, visual representation of fixturing and of the processing.

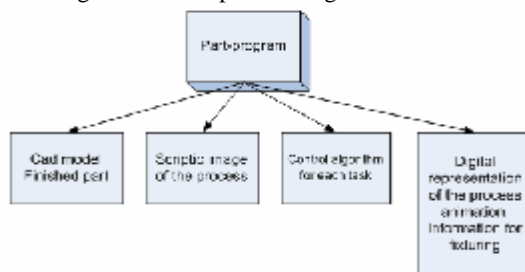


Fig. 7 The part- program

The script image is automatically generated in the visual environment, and for each task it is attached to control software module, responsible for each task control. The machining parameters for the operation are computed in the control module software.

As shown bellow, the script image comprises information concerning the task

successive definitions. For instance, in the example given it is defined Task 1, as the trajectory between point P1 and P2, with the requirements represented by the requirement Rz and tolerance defined by Ts and Ti range. It is also stated characteristic material MatCh and that of the tool Tool1. Also, it is stated auxiliary actions such as fast displacement, starting the coolant, etc.

```

Define P1(X,Y,Z)
Define P2(X,Y,Z)
Define P3(X,Y,Z);
FastPos(X,Y,Z)
ChangeTool(Tool1)
StartCoolant
Task1(P1,P2)(Rz,Ts,Ti,MatCh,Tool1)
Task2(P2,P3)(Rz,Ts,Ti,MatCh,Tool1)
Stopcoolant
FastPos(X,Y,Z)
Changetool(Tool2)
Task1(P3,P4)(Rz,Ts,Ti,MatCh,Tool1).
  
```

4. Conclusions

A new programming approach is proposed in this paper to simplify RMT system development and enhance development ability. To implement this approach, a new control architecture is proposed. This architecture is based on the idea of independent hardware modules, with close loop control feature, controlled directly by a PC.

The programming approach is considering two technological documents: i) machine-program, concerning the machine reconfiguration phase and ii) part-program, concerning the programming of the machine to process a specific part. A visual language architecture is proposed for building the technological documents, allowing a fast programming and an accurate control.

The programming approach proposed can be implemented successfully to the common CNC to offer fast programming for part-program.

The new approach proposed for programming RMT presents several advantages such as: a) the system is universal, intelligent, autonomous and has the ability to adapt to changing, b) it can be used also for programming of other CNC machining, c) it detects collisions of cutting tool, d) it provide the capability to rapid programming at each new machine configuration and offers the capability to a better process control.

The research team is currently working to develop the proposed programming approach and the initial results show the architecture is viable.

Acknowledgement

The authors gratefully acknowledge the financial support of the Romanian Ministry of Education and Research through grant PN_II_ID_653/2007.

References

1. **Cho Myeong-Woo, Kim Gun-Hee, Seo Tae-II, Hong Yeon-Chan, Cheng H. Harry**, *Integrated machining error compensation method using OMM data and modified PNN algorithm*, International Journal of Machine Tools & Manufacture 46 (2006) 1417–1427;
2. **Galan R., Racero J., Eguia I., Garcia J.M.**, *A systematic approach for product families formation in Reconfigurable Manufacturing Systems*, Robotics and Computer-Integrated Manufacturing 23, 2007, 489–502;
3. **Jaspreet Dhupia, Bartosz Powalka, Reuven Katz, A. Galip Ulsoy**, *Dynamics of the arch-type reconfigurable machine tool*, International Journal of Machine Tools & Manufacture 47, 2007, 326–334;
4. **X.W. Xu, Q. He**, *Striving for a total integration of CAD, CAPP, CAM and CNC*, Robotics and Computer-Integrated Manufacturing, 2003;
5. **Jeong Hoon Ko, Won Soo Yun, Dong-Woo Cho**, *Off-line feed rate scheduling using virtual CNC based on an evaluation of cutting performance*, Computer-Aided Design 33, 2003, 383-393
6. **Stanislav Makhanov**, *Optimization and correction of the tool path of the five-axis milling machine Part 1. Spatial optimization*, Mathematics and Computers in Simulation 75, 2007, 210-230;
7. **Stanislav Makhanov**, *Optimization and correction of the tool path of the five-axis milling machine Part 2: Rotations and setup*, Mathematics and Computers in Simulation 75, 2007, 231-250;
8. **Soa B.S., Jungb Y.H., Kurfess T.R., Hwang S.M.**, *5-Axis machining speed enhancement by step length optimization*, Journal of Materials Processing Technology 187–188 75, 2007, 2-5
9. **Gallo M., Guizzi G., Zoppoli V.**, *A methodological approach to develop an integrated simulation system in manufacturing processes*, 6th WSEAS International Conference on System and Simulation and Engineering, 2007;
10. **Paolucci M., Revetria R., Tonelli F.**, *An Agent-based System for Sales and Operations Planning in Manufacturing Supply Chains*, 6th WSEAS International Conference on System and Simulation and Engineering, 2007;
11. **M. Yurtseven**, *Design and performance analysis of a linear quadratic Gaussian controller in a manufacturing process*, 6th WSEAS International Conference on Circuit, System, Electronic, Control and Signal Processing, 2007;
12. **OMAC API Work Group** OMAC API set, <http://www.isd.mel.nist.gov/projects/omacapi/>, 1999
13. **OSACA** *Open system architecture controls within automation systems*. Final report. <http://www.osaca.org/>, 1996;
14. **Wang S, Shin KG**, *Constructing reconfigurable software for machine control systems*. IEEE Trans Robot Autom 18(4):475–486, 2002;
15. **Wei Li & Yadong Liu & Kazuo Yamazaki & Makoto Fujisima & Masahiko Mori**, *The design of a NURBS pre-interpolator for five-axis machining*, Int J Adv Manuf Technol, 2008;
16. **Epureanu AL, Marin F.B., Marinescu V., Banu M., Constantin I.**, *Manufacturing machines – a holonic approach*, Proceeding ACMOS 08 ISBN:978-960-6766-63-3, ISSN 1790-5117, 2008;
17. **Epureanu A., Marin F.B., Marinescu V., Banu M., Constantin I.**, *Optimal feedrate scheduling for a reconfigurable lathe*, Proceeding ACMOS 08 ISBN:978-960-6766-63-3, ISSN 1790-5117, 2008.

Conceperea unui limbaj de nivel înalt pentru programarea mașinilor reconfigurabile și a programului de prelucrare necesar noilor generații de mașini-unelte reconfigurabile

Rezumat

Lucrarea propune o noua abordare în ceea ce privește programarea RMT-urilor (Reconfigurable Manufacturing Tool). Abordarea se bazează pe o nouă arhitectură de control caracterizată printr-un înalt grad de reconfigurabilitate. Este propusă ideea a două documente tehnologice i) programul-mașina, care corespunde fazei de reconfigurare și ii) programul-piesa care corespunde fazei de procesare a unui semifabricat. Un limbaj vizual de nivel înalt este propus pentru a genera cele două documente, permitând o programare rapidă și un control precis al procesului.

Conception d'un langage de haut niveau pour la programmation de l'usinage de reconfiguration et de pièce de machine consacré nouvelle aux machines-outils la génération reconfigurable

Résumé

Ce document propose une nouvelle approche pour l'issue de programmation de RMT (outil Reconfigurable de fabrication). L'approche est basée sur une nouvelle architecture de commande caractérisée par la reconfigurabilité élevée. On lui propose l'idée du machine-programme technologique de deux documents i), au sujet de la phase de reconfiguration de machine et ii) du partie-programme, au sujet de la programmation de la machine de traiter une cloison spécifique. On propose l'architecture d'une lang visuelle pour les deux documents technologiques, permettant une programmation rapide et une commande précise.