

APPLYING THE CONCEPT OF SOFTWARE PRODUCT LINES TO CONSTRUCT LOW-COST CNC

⁺Guillermo Jiménez, ⁺⁺Miguel de J. Ramírez, ⁺⁺⁺Arturo Molina

⁺*Informatics Research Center*

⁺⁺*Mechatronics and Automation Department*

⁺⁺⁺*Engineering and Architecture Division*

Instituto Tecnológico y de Estudios Superiores de Monterrey

Av. E. Garza Sada 2501 Sur, C.P. 64849, Monterrey, Nuevo León, México

Tel. +52(81)83582000 x5485, Fax +52(81)83284123

{guillermo.jimenez@itesm.mx, miguel.ramirez@itesm.mx, armolina@itesm.mx}

Abstract: The manufacturing industry in developing countries is being challenged to compete in unequal circumstances with larger multinational enterprises. Most of the manufacturing floor in developing nations is characterized by the use of manual machine-tools, which are unable to produce work products with the characteristics and quality that can only be achieved by automatic machines. Here we propose to retrofit manual machines following a low cost approach for their automatization. The approach consists in using low-cost actuators, mechanical devices, electronic components and the controlling software is built by a software product lines approach.

Keywords: computer numerical control, software product lines, retrofit, machine-tools, low cost automation.

1. INTRODUCTION

The Research Chair on Mechatronics of the Monterrey Institute of Technology (ITESM) has developed the concept of Universal Numerical Control (UNC) based on open system concept and PC technology to develop a CNC architecture to reach low cost automation by retrofit manual machine-tools. A retrofitted machine tool is a manual machine-tool, in which many of the manual levers are substituted by actuators controlled by a computer numerical control (CNC) system. The instructions to manufacture a work piece are provided by a numeric control (NC) program.

The rationale of UNC project is related with the fact that in countries like Mexico, small and medium metal-working industries work with manual machine-tools and rarely have access to computer numerical control (CNC) machine-tools (Ramírez, 2001); only well established companies have access to automated machine-tools due to its cost. The research group has identified that Mexican small and medium metal-working industry has been facing major problems in international markets due to the lack of CNC technology which is the basis for competition in this sector (Ramírez, 2004).

Therefore an area of opportunity has been identified to develop national technology in the

area of CNC using low cost PCs and software based open system architectures implemented as software product lines. The concept of Universal Numerical Control aims to tackle two important issues: to provide the Mexican industry with low cost CNC technology and to explore the new concepts behind the development of open CNC architectures and software product lines.

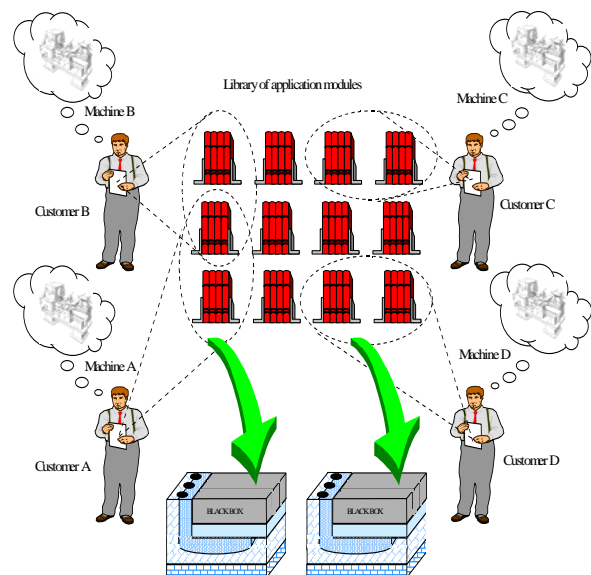


Figure 1. Universal Numerical Control concept.

The Figure 1 shows the concept of UNC in order to retrofit different machine-tools; library of “Black Box” software modules which can be integrated to the UNC reference architecture to produce CNC controllers. Customers can choose the specific software modules to configure their own CNC controller (Ramírez, 2001).

To achieve the open system control concept, the internal architecture of the control system needs to be supported by a platform which hides the hardware-specific details from the software components and define a flexible way of communication between the software components. The control system is subdivided into several software modules interacting through a defined application programming interface API (Pritschow et al, 2001). The open architecture is used to design and implement systems formed from different combinations of software modules using a plug and play approach on the architecture to reach the needed functionality (Ramírez, 2002).

A reference architecture is a conceptual model that specifies rules and methods for the integration of components into a system by means of standardized interfaces to defining a structure that permits different developers to build systems with characteristics specified by the architecture (OSACA, 2001). The UNC project has created a reference architecture to give flexibility in order to adapt the control platform to deal with different types of machine-tools like lathes, milling machines and drilling machines. These machines use different types of actuators (step motors, DC motors, AC motors and others), a large variety of sensors (proximity, encoders, tachometers and others), mechanicals components (ballscrew, brackets, pulleys and others), math interpolators, and human-machine interfaces.

UNC reference architecture is based on the ESPRIT project called OSACA (Open System Architecture for Controls within Automation systems). OSACA is an European initiative to define a vendor-neutral open architecture controller in order to improve the competitiveness and flexibility of suppliers and users of control systems: machine tool builders, control vendors and end-users. The justification of OSACA emerges from the need of European countries of standardizing their CNC industry to compete with Japanese and American industries. The UNC reference architecture follows this open architecture frame and maps it to the needs of developing countries, which could benefit from the availability of an approach to have access to a low cost CNC technology by retrofitting manual machine-tools to upgrade the technology base of their SMEs.

Figure 2 shows the UNC reference architecture which is a multi-layer system separating the hardware and software platforms that provide services to the software functional units (FU's). The hardware system consists of the hardware layer which includes electronic components that make part of the controller (motor drivers, sensors, etc.), the circuits of the computer and its peripherals, such as I/O and network boards. The software system contains the operative system layer, communication layer and the application programming interface (API).

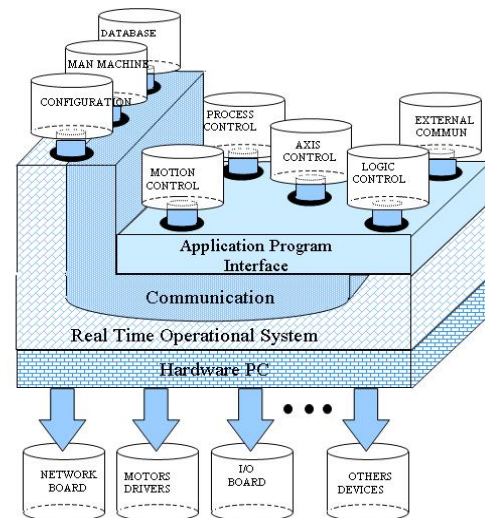


Figure 2. UNC Reference Architecture.

Five FU's were identified as application software units and three FU's as system software units. Figure 3 shows the information flow between the main FU's in a CNC environment.

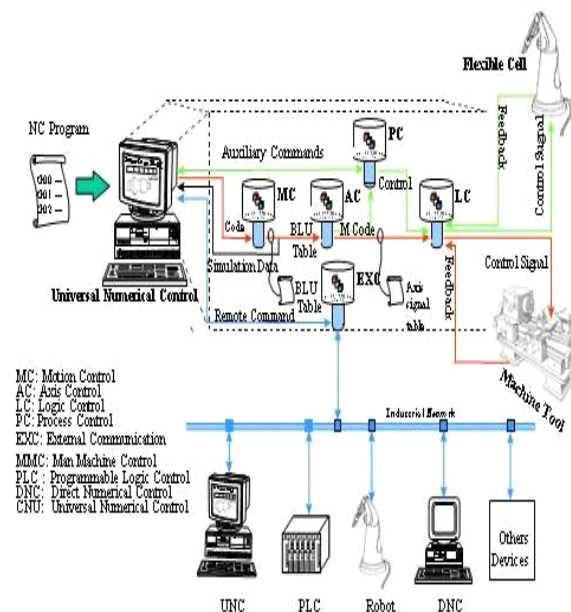


Figure 3. Information flow between FU's.

A short definition of each FU is included below (Noriler, 2003) (Ramírez et al., 2004):

- a) Man-Machine Control (MMC): Is the user interface to allow the user control the operation of the machine.
- b) Motion Control (MC): enables the machine to produce relative motion by calculating the trajectories that must be executed using math interpolator algorithms.
- c) Axis Control (AC): includes all the means necessary for activating the axis to execute movement. It is responsible for generating the signal tables and sends it to each axis of the machine-tool.
- d) Process Control (PC): represents the auxiliary systems of the machine, such as spindle, coolant, lubrication system, etc. They are also responsible for the data management and processing of the auxiliary systems they represent.
- e) Logic Control (LC): responsible for the operation of the actuators and data taking of the sensors built in the machine. They should also ensure the consistence of the operations and the data taken.
- f) External Communication (EXC): provides the communication of the system with the exterior by an industrial network of communication or load programs from a computer (direct numerical control, DNC).
- g) Simulation (SIM): permits machining simulation.

As mentioned before, besides the previously described FU's, there are the database (DB) FU and the configuration (CONF) FU. Database stores information such as tool data, process parameters, etc.; and the configuration FU is responsible of instantiating and establishing communication connections between FU's.

2. PRODUCT LINE OF CNC SYSTEMS

As described in the previous section, the control software of different machine-tools has a number of similarities and only minimal differences. For instance, every automatic machine-tool have sensors and actuators, whose number is different for different machine-tools, but still the reference architecture of the control software for different machine tools have remarkable similarities. The main differences are in the specific actuators, number of axis, interpolator algorithms, control algorithms and man-machine interface.

The Product Line Engineering (PLE) concept, consists in developing software architecture based infrastructures that could be used to produce a set of applications that share the product line

architecture and valid combinations of software elements to produce application systems (Weiss and Lai, 1999).

As mentioned in the early section, the main goal of our research is to produce low-cost controllers for retrofitted machine tools. By implementing the control software in low cost PCs, instead of special purpose computers or PLCs, the total cost of the controlling element could be reduced. Clearly, a product-line based approach is a good candidate for the control software. The following section describes how such possibility could be implemented for CNC systems.

The development of a software product line involves two steps. It is first necessary to analyze the domain to understand the functionalities and their interdependencies, in order to design the software architecture. Because the machine-tool domain is characterized by different physical components, both mechanical and electronic, the resulting model representing these components should include these elements and their dependencies. For instance, it is necessary to know if a particular data acquisition card supports both digital and analog signals, and its number of inputs and outputs, because not every data acquisition card could be appropriate for every type of machine-tool. It is also necessary to know the characteristics of the actuators, thus they can be correlated to the possibilities of the data acquisition card.

To represent the characteristics of the CNC domain and the particular hardware elements that should be controlled, we use a feature diagram (Kang et. al, 1990). A feature diagram is a tree of nodes, in which every node represents a machine-tool element, and its relationship to other elements in the machine-tool, or the possible instantiation of the element (i.e., the possible data acquisition card that could be used for a specific set of actuator). Also, the feature diagram should reflect constraints showing invalid combinations of hardware or software elements. Figure 4 shows the particular notation for a feature diagram emphasizing our notation.

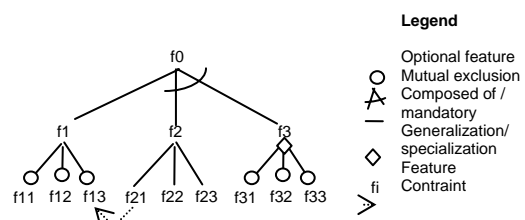


Figure 4. Feature Diagram Notation.

Using our conventions described in Figure 4, a partial feature diagram for CNC systems is presented in Figure 5. CNC systems have two general features: a motion generator and a motion control. Motion generator interprets the descriptions of a NC program specifying the operations for the actuators in the machine-tools (e.g., coolant on/off, spindle speed and direction, target positions of the cutting tool, etc.).

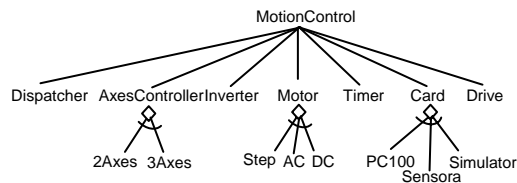


Figure 5. Feature diagram of a motion control.

The motion generator translates the NC program instructions to a unified representation thus the task of the motion controller could be simplified. This is important because the motion controller should coordinate the operation of the machine tool elements at real-time to warranty the expected quality of the work product.

Figure 5 shows the motion control, describing the possible combinations of hardware and control elements in the resulting CNC to warranty that the machining operation is conducted in a correct way, using the appropriate controller. For instance, the motion controller differentiates on how signals are sent to a stepper motor or to DC motor, as their operation is quite different, and the appropriate controlling software should be included in the motion controller.

A step in the product line approach is still missing, that is: how a selection of features for a particular CNC could be translated to a specific application system of a CNC. For this, the reference software architecture is necessary (according to the methodology of software product lines). It is possible to use different architectures; we propose the use of a hierarchical architecture based on the reference architecture mentioned in early sections, in which every software element transforms its input into an output that could be feed to a software element below it. A generic hierarchical architecture for a CNC product line is depicted in Figure 6. The architecture is generic in that the lower software element at any layer is a parameter of its upper layer element. The approach of generic hierarchical software architectures has been applied to produce sophisticated data structures as compositions of software elements (Nierstrasz and Meijler, 1995), whose performance have demonstrated to be better than applications using

different architectures (Jiménez and Batory, 1997).

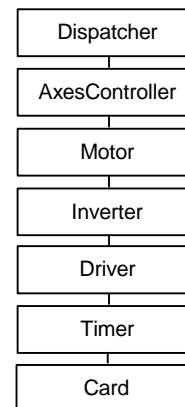


Figure 6. Hierarchical diagram of a CNC control.

Following the recommendations of the software PLE approach, the final step is to design a tool to help users to produce application systems as compositions of the software elements described by the generic architecture and verify that the intended compositions are valid.

The approach followed is to use a software wizard to assist users in configuring specific CNCs. Software wizards have a long history as graphic tools to walk the user to specify intended features of an application. Many of the installers of software applications provide a wizard to free the user of executing a set of complex commands to configure applications.

A wizard for CNC configuration should simplify the specification of features that particular CNC's should support. However, a wizard for a product line of CNC should extend the support for providing users with feedback of the intended configurations. If a CNC configuration is valid, a description of the characteristics of the CNC produced would help to verify that user intentions correspond to the compositions produced by the wizard. If an invalid combination of features is provided to the wizard, users would receive feedback about why the provided specification was invalid, thus correction can be done.

Figure 7 shows the user interface of the configuration wizard for CNC specification. As could be seen, many of the features defined by the feature diagram, depicted in Figure 5, are not shown in the user interface. This is because mandatory features (i.e., features that are always part of applications) don't need to be specified and are automatically added by the wizard.

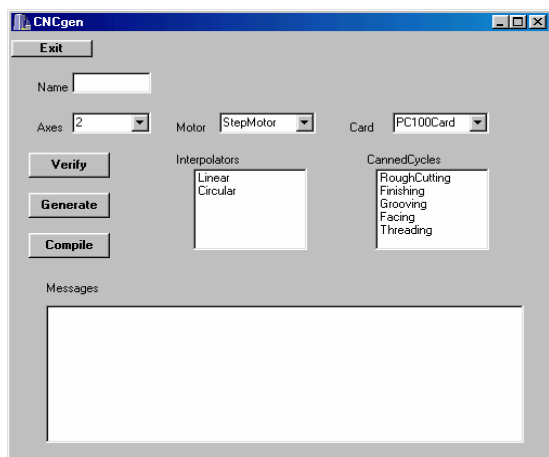


Figure 7. GUI for configuration wizard.

3. RESULTS

The CNU project allow the retrofit of machine-tools by the addition of actuators, sensors, mechanical devices and PC with CNU control software, data acquisition and network cards. Figure 8 shows the implementation concept of CNU which is made dependent of equipment investment.

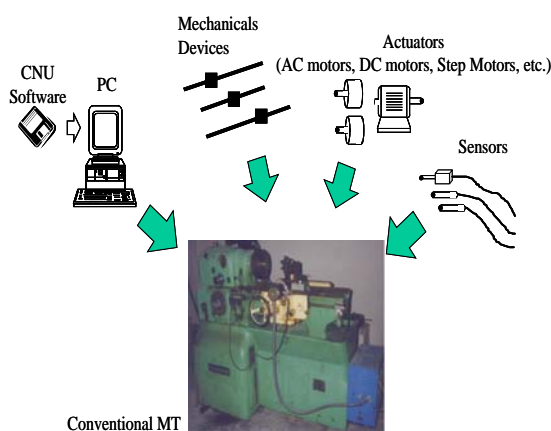


Figure 8. Implementation of the project

The approach described has been used to produce CNC for a retrofitted two axis lathe. The CNU prototype was made with a Monarch lathe 10EF built on 1941 with 12.5" of swing and 20" of distance between centers. Two ballscrew of 18" length for X axis and 36" length for Z axis both with 0.2" of pitch and 3/4" of diameter were mounted. A Mitsubishi FREQROL-Z200 inverter was connected to 4 kW and 220 Volts three-phase motor. Two 625 oz-in stepper motors with encoder and drivers with 50,000 microstep of resolution. A Sensoray 626 PCI Multifunction data acquisition board was inserted in a slot of the

PC. The board have I/O channels, analog inputs and analog outputs.

The tests executed on the lathe shown that machined pieces satisfied the qualities of a machine-tool controller. The total cost of the resulting machine was significantly lower than a commercial CNC machine-tool or CNC retrofit kit available on the market.

The software system was developed in C++ running on the real time QNX operating system. The configuration wizard helped to configure the CNC system with the modules corresponding to used hardware. The wizard simplified configuration changes by using a library of software modules. CNC systems with different options could be configured; for example, choosing the number of axes, the DAQ card available, the motor type for feed, the inverter to drive the spindle motor, the interpolator used and the canned cycles available.

4. CONCLUSIONS

An approach based on software product lines was applied to the production of CNC systems for retrofitted machine-tools. The hard real-time requirements of CNC were demonstrated by retrofitting the CNC for a two-axis lathe.

Finally the research group considers the CNU concept an excellent software based technology, by its low cost and high technical flexibility, which will impact the development of the small and medium national metal-working industry. With the CNU, the user can reach a capitalization and know-how about CNC technology so that latter it will be possible to buy larger commercial CNC machine-tools.

The next steps are to retrofit other kind of machine-tools in small and medium metalworking enterprises. Also, the research group has begun the retrofitting of lathes and milling machines that will be used to train students on CNC concepts and programming.

5. ACKNOWLEDGEMENTS

The research reported in this paper is part of a Research Chair in Mechatronics of ITESM titled "Design, Manufacturing and Integration of Reconfigurable and Intelligent Machines". The authors wish to acknowledge the support of this grant in the development of this research.

6. REFERENCES

- Jiménez, G. and Batory, D. (1997). Memory Simulators and Software Generators. *Proceedings of the Symposium on Software Reusability*, Boston, Mass.
- Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., and Peterson, A.S. (1990). Feature-Oriented Domain Analysis (FODA) Feasibility Study. *Technical Report CMU/SEI-90-TR-21, Software Engineering Institute*, Carnegie Mellon University.
- Nierstrasz, O. and Meijler, T. (1995). Research Directions in Software Composition. *ACM Computing Surveys*. 27(2):262-264.
- Noriler, M.A. (2003). Design Methodology for Universal Numerical Control: Application to EMCO Compact 5 Lathe. *College Thesis*. Universidade Federal do Santa Catarina, Brasil.
- OSACA Association. (2001). OSACA Handbook. *ESPRIT Project III*. (www.osaca.org).
- Pritschow G., Altintas Y., Jovane, F., Koren Y., Mitsuishi, M., Takata, S., Van Brussel, H., Weck, M., and Yamazaki, K. (2001). Open Controller Architecture – Past, Present and Future-. *Annals of CIRP*.
- Ramírez, M. de J., (2001). Software based Computer Numerical Controller for Low Cost Automation in Small and Medium sized Metal-Processing Enterprises in Developing Countries. *Memories of the VI IFAC Symposium on Cost Oriented Automation*, Berlin, German.
- Ramírez, M. de J., Molina, A., Jiménez, G., Noriler, A., (2004). Design methodology for cnc applications based on open systems. *Preprints 7th IFAC Symposium on Cost Oriented Automation COA 2004*, Ottawa, Canada. pp. 153-158.
- Weiss, D. M. and Lai C. T. R. (1999), *Software Product-Line Engineering*, Addison-Wesley.