Reengineering a Computerized Numerical Control Towards Object-Oriented

F. Butera*, B. Fontanella*, P. Nesi!, M. Perfetti*

*ELEXA S.r.L., CB-Ferrari Group

Via delle Panche 38/A, 50141 Florence, Italy, elexa@mbox.vol.it,

!Department of Systems and Informatics, Faculty of Engineering, University of Florence

Via S. Marta 3, I 50139 Florence, Italy, nesi@ingfi1.ing.unifi.it

Abstract

The experience of reengineering a Computerized Numerical Control, CNC, is presented. In the reengineering, several new functionalities and the upgrading towards the object-oriented were included. The reengineering process was maintained under control in order to complete the process within the deadline with success. Some technical and economical lessons learned that can be useful for helping the readers in solving similar problems are discussed.

1 Introduction

The main component of factory automation is the CNC (Computerized Numerical Control), which provides a set of functionalities for the management of Machine Tools, MTs. The adoption of a CNC for controlling MTs, instead of using classical PLCs (Programmable Logic Controllers), provides support for implementing complex strategies and thus for building complex MTs such as Milling Machines, MMs, for templates, for rapid prototyping, and for producing models, etc. These are strongly complex and completely reconfigurable machines. The production of these machines is very close to a custom process, since the purchaser asks for a machine requiring specific: tables, axes, actuators, measuring systems, etc.

In the last years, with the reengineering of most of the applications, the passage from procedural to object-oriented approach has been also planned. This, because the object-oriented promises to produce software with high degrees of reusability and maintainability, satisfying in a certain measure also quality characteristics [1], [4], [5], [6], [3]. Thus, the object-oriented paradigm (OOP) is presently considered the one which best guarantees the investment for the renewal. The above mentioned benefits of the OOP are not obviously automatically guaranteed by the simple adoption of an object-oriented programming language, though this is a common opinion.

In this paper, the reengineering of a CNC for machine tools is presented. The starting version of the CNC was based on bi-processor architecture in which the user interface was built in MS-DOS. The reengineering started from the analysis of the requirements coming from consumers. The reengineering of the old version was planned on the bases of a preliminary study to perform a direct experience in modeling the problem domain of CNC according to the object-oriented paradigm. On the bases of the object-oriented models and using the support for building applications under Windows NT the new version of ELEXA-CNC has been built.

2 The Old Application and the New Requirements

The old version of the CNC reengineered was implemented and improved in several years of work by many people. ELEXA-CNC is based on biprocessor architecture of i486 or Pentium. Each processor is supported by a board very similar to a Personal Computer. The main functionalities of the old version of CNC are: (i) interpretation of ISO programs; (ii) generation of micropoints and controlling axes; (iii) evaluation and generation of the control values for machine actuators; (iv) execution of logical rules; (v) a user interface; (vi) editing of ISO programs; (vii) simulation of tool path or ISO program on the screen; (viii) configuration of the machine tool; (ix) monitoring of errors and alarms.

The above functionalities are assigned at the two processors of the ELEXA-CNC (see Fig.1). One processor, called CN (Control Numerical) performs (i), (ii), (iii), and (iv). The second processor, called UCNC (User CNC), supports the first providing ISO programs and performs (i) for the simulator (vii), and (vi), (viii), (ix). The two processors exchange information in real-time by means of a dedicated shared memory. In the MS-DOS version, all control pan-

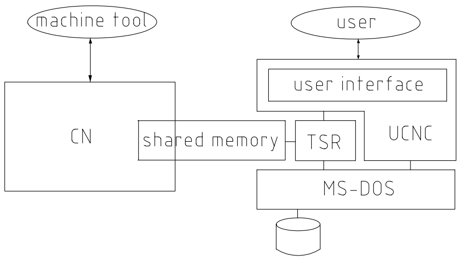

Figure 1. A simplified view of the architecture of the MS-DOS version of ELEXA-CNC.

Figure 2. The main screen of the MS-DOS version of ELEXA-CNC.

els were realized under MS-DOS without the adoption of a windowing system. See Fig.2, where the values related to the position of axes can be observed on the left, while on the right several variables for reporting the status of the MM during the execution of the working program.

2.1 Reengineering Requirements

Before the starting of the reengineering process, several interviews were performed for identifying the reasons for its execution and the new needs of the end-users. After an analysis of the interviews performed at the end-users the main issues to be renovated were:

– the user interface: (i) to adopt a window-based user interface; (ii) to allow the configuration of the user interface;

– the operating system of UCNC to allow: (i) the concurrent access to network facilities; (ii) the integration of ELEXA-CNCs into CIM policies; (iii)the integration between the CNC and CAD/CAM area; (iv) the execution of working program concurrently with the other operations;

– the simulator of working programs on UCNC: (i) to increase performance; (ii) to allow the synchronization with the execution of instructions on the CN and the tracing of the instructions on the editor of working programs; (iii) to improve capabilities of the editor in managing long files; (iv) to allow the simulation concurrently with the monitoring of CN values;

– the error, alarm and help manager: (i) to allow selective and programmable logging of errors and alarms; (ii) to allow the interactive visualization of log file; (iii) to provide an on-line Help;

– the configurator: (i) to provide a user friendly interface; (ii) to increase the types of actuators; (iii) to increase the types of sensors and measuring equipments; (iv) to increase flexibility;

2.2 Architectural Aspects: Reengineering Process

Comparing the old MS-DOS version and the list of the new functionalities, it was evident that several functionalities of the old version could be reused for building the new version. In particular, it was noticed that

– the changes required mainly involved the UCNC and only marginally the CN part. The CN is about the 40 % of the whole application which in turn is comprised of CN, UCNC with all complementary programs of support;

– the procedure implementing mathematical algorithms could be totally reused. These were mainly located in the CN part, they were the 70 % of the CN. Excluding the simulator, in the UCNC part, the mathematical procedures are only the 10 % of the total code.

On the contrary, a great effort was evaluated for: defining the concurrent architecture; implementing the user interface; allowing the configuration of the user interface; improving the machine tool configurator; improving the simulator; building a strongly updated version of the manager for errors and alarms; building the help manager and assistant.

After the analysis of the previous mentioned issues, the only solution for the reengineering problem was to remake the application by reusing the code and the problem domain experience. It has been decided to divide the process in two phases. In the first phases only the UCNC has been reengineered, it is the 60 % of the system. Thus, the CN has been maintained in the old version up to the complete reengineering of the UCNC.

It was decided to produce the new version by migrating towards the object-oriented paradigm and using C++ on Windows NT platform. Thus, the new architecture presents additional modules with respect to the MS-DOS version (see Fig.3).

Considering only UCNC, it was constituted by source code in C: 500 Kbs for the configurator, 470 Kbs for the UCNC, 550 Kbs for headers files, 450 Kbs for the simulator, 750 Kbs of Assembly code. Thus, a total of 2350 Kbytes of code which result in 39600 Lines of Code, LOC, for the C part and about 16000 LOC for the Assembly part.

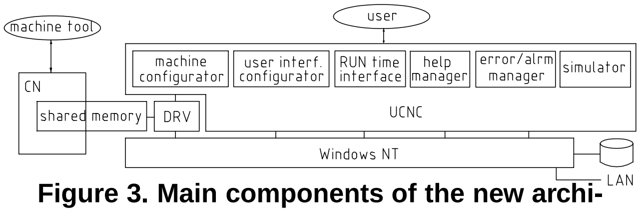

Figure 3. Main components of the new architecture of the ELEXA-CNC.

3 The Object-Oriented Computerized Numerical Control

In order to pass from a procedural version of a program/application to an object-oriented version, a process of re-analysis is mandatory. This process leads to the identification of the classes that model the application domain and their organization [1], [7], [2].

3.1 Early Phases of the Reengineering Process

The reengineering of the old version started in 1993 with a preliminary analysis of the application domain for verifying the applicability of the OOP and evaluating its costs. That work produced an object-oriented model of the application domain (70 classes, in C++). On this basis, the decision to apply the object-oriented to the whole area was taken, with a special attention to define a machine model: (a) suitable to be reused for configuring, executing, simulating the machine behavior; (b) independent of the user interface.

In the 1995, the real reengineering process started for producing an object-oriented version of UCNC on Windows NT. The implementation of the above discussed new activities in the selected platform was evaluated for total 51 man-months considering analysis, design, code, test, management and the 50 % of the documentation (see section 2.2): architecture (7 mms); driver for communicating with the CN (2 mms); user interface (12 mms); configuration of the GUI (6 mms); machine tool configurator (7 mms); simulator (8 mms); manager for errors and alarms (4 mms); help manager and assistant (3 mms).

The effort of each subsystem/functionality was estimated by using the experience and predictive metrics [6]. Please note that 12 man-months were planned for realizing the new user interface under Windows NT by starting from the MS-DOS version. The MS-DOS version had 25 different screens while the new object-oriented UCNC provides 46 different windows. A relevant reuse was performed in (5) and in (6), where several mathematical procedures were reused. According to the above decision and on the bases of our experiences, the reusing of about the: 20 % of the analysis, 25 % of the design, 10 % of the code, and 80 % of the test cases, were predicted and supposed before the beginning the reengineering of the UCNC. The documentation could be reused for at least a 40 % since most of the screens of the user interface have to present the same information.

The phase of analysis was to be quite completely remade since the OOP changes substantially the point of view. Only the knowledge of the application domain could be reused, saving about the 10 % of the effort. The phase of analysis of the problem domain was totally reused from the preliminary study about the object-oriented model of CNC domain. The design could to be partially reused since the screens and the main subsystems of the application were already identified. The reuse of coding was limited to the inclusion of mathematical procedures and pieces of code for data conversions.

3.2 A Short Overview of the New User Interface Capabilities

The application presents a menu bar on which several different menus can be installed according to the context, see Fig.4. Below the menu, a set of buttons (a line of big buttons over a line a small buttons) is available. The large buttons are used for activating small windows, that are called parts. Parts are used for monitoring values about the machine tool at runtime. During the configuration of the user interface, the user can arrange parts in any position and size for composing the screen. These can be saved into the GUI configuration. Below the parts, a set of function keys is available. Pressing the function keys, as well as clicking them with the mouse, it is possible to recall other set of parts for covering the screen. The composition of parts in the screens and their shape can be defined during the configuration of the user interface. Thus, the screen can be associated with function keys that can recall also other sets of function keys or send direct commands. In this way, the whole set of screens of a CNC-based application can be built. This is very useful for customizing the user interface according to the user of machine builder requirements.

3.3 Some Quantitative Data and conclusions

A great part of the work performed for reengineering the UCNC will be totally reused by the reengineering process of the CN part. This will be also possible since, during the reengineering of the UCNC, a special policy of design for reuse has been applied.

At the 70 classes used for modeling the object-oriented domain framework for CNCs several other classes have been added. Thus, 28 classes for the ISO interpreter and editor; 87 for user interface; 33 for simulator; 7 for user interface configuration; 4 for architectural and multitasking; 3 for communications; 6 for error and alarms manager; etc.

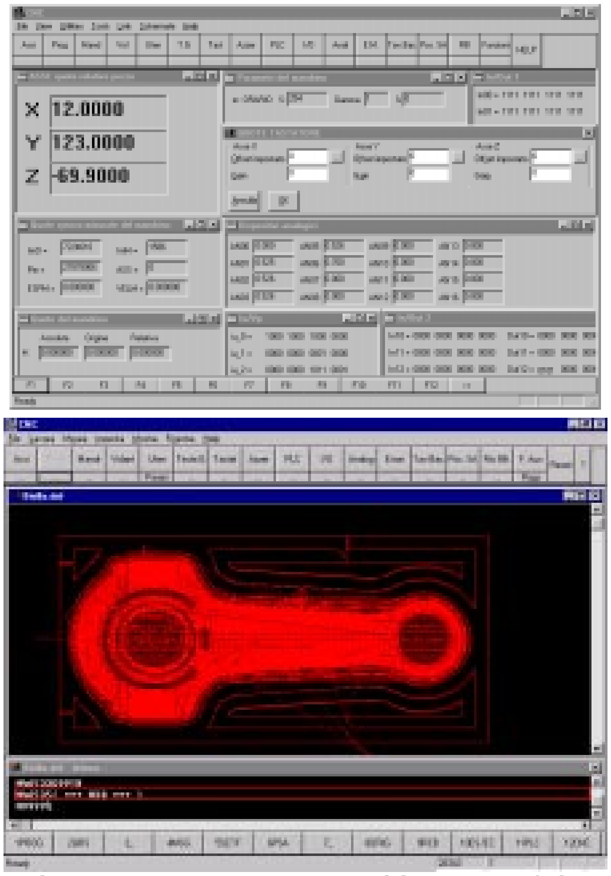

Figure 4. Screens composed by parts of the ELEXA-CNC (please note the simulator).

Thus, the application resulted in 4.5 Mbs of code, for about 10000 LOC.

Some of the above mentioned supports have been implemented as the fundamental framework/clusters of classes for realizing the global framework. Thus, even the costs for reengineering the CN parts will be reduced since all classes for modeling the machine have been already implemented. To these classes some functionalities should be added but they have been evaluated to be the 10 % of the total complexity of the class.

In Fig.5, the comparison between the predicted effort and the effective effort needed for the main functionalities/subsystems of the ELEXA-CNC is reported. For the occurrence of some technical problems due to the complexity of the system and to the parallelization of work among the people the actual effort resulted of 54.9 man-months. This has to be considered a success since only an error of -7.6 % was made with respect to the prediction made at the beginning of 1997 for getting the deadline of Hannover Fair in September 1997.

The predictions about the reuse were quite precise. The development of UCNC was maintained under control by using assessing the system every month [6]. The quantitative monitoring of the development has allowed the identification of the above-mentioned extra-effort since March 1997. In fact, in that period the planned Gantt and actual Gantt began to present a relevant misalignment (more than two weeks).

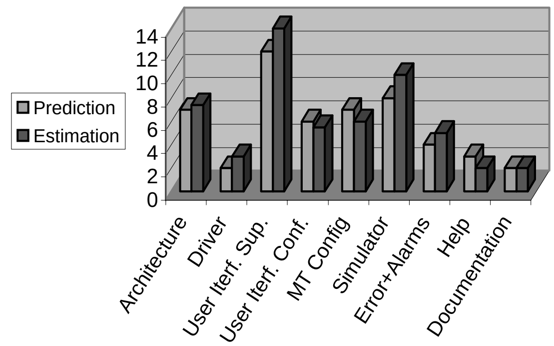

Figure 5. Predicted and effective effort for the main features of ELEXA-CNC.

The object-oriented framework defined for modeling the CNC domain can be profitably used as a basis of several other applications: CNCs for robots, CNC for measurement machines, CNC for pipeline of productions, etc. It is currently used by ICCOC (Integrating CAD/CAM Operations into CNC-Based Machines) project (ESPRIT HPCN TETRApc-TTN, partners: ELEXA, University of Florence, TECMA s.r.l.) for the analysis of the integration strategies.

References

[1] G. Booch. *Object-Oriented Design with Applications*. The Benjamin/Cummings Publishing Company, California, USA, 1994.

[2] A. Fantechi, P. Nesi, and E. Somma. Object-oriented analysis of COBOL. In *Proc. of 1st Euromicro Conference on Software Maintenance and Reengineering*, Berlin, Germany, 1997.

[3] M. Lorenz and J. Kidd. *Object-Oriented Software Metrics, A Practical Guide*. PTR Prentice Hall, New Jersey, 1994.

[4] P. Nesi. *Objective Software Quality, Proc. of Objective Quality 1995, 2nd Syposium on Software Quality Techniques and Acquisition Criteria*. LNCS N.926, 1995.

[5] P. Nesi and M. Campanai. Metric framework for object-oriented real-time systems specification languages. *The Journal of Systems and Software*, 34:43–65, 1996.

[6] P. Nesi and T. Querci. Effort estimation and prediction of object-oriented systems. *The Journal of Systems and Software*, 1998.

[7] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall International, 1991.