

# Group technology for automated generation of machine controller code

O. Hornyák\* and G. Sáfrány\*\*

\* University of Miskolc/Department of Information Engineering, Miskolc, Hungary

\*\* evoPro Ltd., Budapest, Hungary  
[hornyak@ait.iit.uni-miskolc.hu](mailto:hornyak@ait.iit.uni-miskolc.hu),  
[safrazy.gabor.ext@siemens.com](mailto:safrazy.gabor.ext@siemens.com)

**Abstract**—This paper focuses on the application of the Group Technology (GT) concept in the case of producing controller software for manufacturing machines. An overview on the modular machines is given. The paper presents a new concept for the development of the controllers' software. The advantages of GT are presented. Two grouping approaches are detailed: the bottom-up approach and the top-down approach. A Genetic Algorithm Clustering is presented for the automatic grouping.

## I. INTRODUCTION

The design, assembly, starting up, monitoring and maintenance of industrial manufacturing devices have become very complex tasks over the time. According to the predictions for the future the complexity will not decrease. Simultaneously, the demands and expectations against the quality of the products and processes have been growing. It is a well known fact that the cost price is an important factor in today's competitive manufacturing.

Another important factor is the lead-in time spent on the aforementioned activities. Engineers need to revise the activities required for producing a manufacturing design to be able to meet today's needs and to do a step towards future manufacturing devices.

The manufacturing machines producers need to develop the code used by their controllers. In case of process control the Programmable Logic Controller (PLC) based solutions are very widespread as the controlling algorithms are stored in a program.

At the moment the engineering tools used for creating such controllers code do not utilize the opportunities of Group Technology (GT), i.e. the reorganization, grouping of the similar hardware and software modules, and creating developer databases. So that the engineering tools offer only a static help in the software development process. Industrial experiences show the fact, that highly qualified engineers are required for confident working with such tools.

Industrial experts urge modularization. This paper provides a small contribution to that work.

## II. OVERVIEW ON MODULAR MACHINES

### A. Modular machines

The design of manufacturing machines – both in the hardware and software aspects – is undergoing a sensible change. The solutions which offer customizations required by the merchant at the designing phase receive attention of the engineers.

- The use modular building techniques – including mechanical parts, standardized communication, standard software modules, and generation of software modules – have been increased.
- It is expedient to use automated tools to develop such a customer-specific controller code which is very often includes motion control modules as well.

Automated controller creation tools have the advantage of:

- increasing the preciosity of repeated tasks,
- reduces the dependency on engineers experience,
- improves quality of the process and product documentation,
- reduces the preparation time of tendering,
- improving the flexibility of the preparation of the engineering work.

### B. Automated generation of project variants

Besides the modular design of the manufacturing machines there is an increased need for implementing methods and tools by which the automated generation of the controller algorithms and project databases can be executed, especially without an expert knowledge of the system. It is not necessary for the involved engineers to have an overall knowledge about the controller details. However a generic change request or a call for a tender may require prompt action to develop the required machines, including the controller's software. Those machines are made to order or engineered to order, which means high level of custom needs. To give a high service level for such customers there should be methods and tools for non expert engineers, marketing and sales staff which enables:

This work was supported in part by the Production Information Research Team of Department of Information Engineering at University of Miskolc. The financial support is acknowledged.

- to combine the requested product (i.e. the generation of the product descriptor database, the automated generation of the controller code, the feasibility checks and availability of the required stock);
- to generate some software modules based on the existing project code library;
- to use an easy to understand, non technical workflow description language;
- to support the developer engineers in executing their repeated tasks;
- to support open source development.

C. Group technology

Group Technology is a management theory based on the principle that similar things should be done similarly. Products requiring similar operations, having similar attributes or needing common set of resources are grouped into product families [3]. A key factor is the coding, which is an activity to create an - usually - alphanumeric key that describes the product and its attributes. The code must be exact and unambiguous. The literature identifies three kinds of codes:

- monocode (hierarchical code), where each character is restricted by the previous character,
- polycode (attribute), where each character has a its own piece of information,
- hybrid (mixed code).

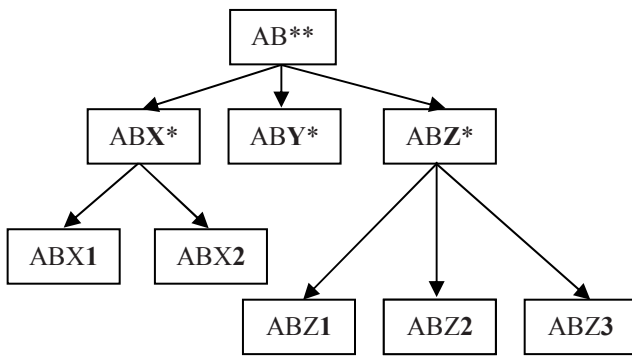


Figure 1. Monocode.

Table 1 Polycode

CODE POSITION	ATTRIBUTE	VALUE OF THE SET			
1	ATTIB. A	1	2	3	...
2	ATTIB. B	A	B	C	...
3	ATTRIB C	NONE	ALL	..	

Possible codes: (1, A, All); (2, C, None); (3, B, All)

III. FEASIBILITY OF GROUP TECHNOLOGY FOR MANUFACTURING MACHINE CONTROLLERS

The development process of creating a new manufacturing machine is a complex task. It is expedient to use the experience from former machine development projects. It is not an acceptable approach which does requires rewriting of an existing functional block of the controller software. Engineers need to split complex projects into small modules which can operate alone. Complex systems need to build from these bricks according to predefined rules.

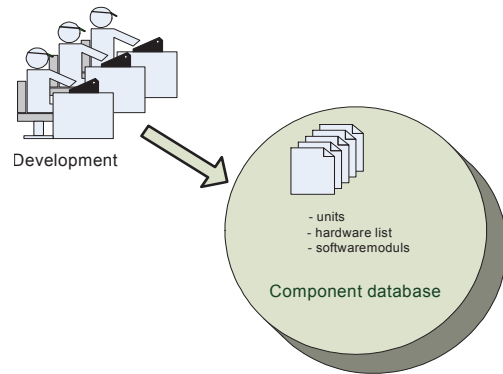


Figure 2. Development process of a new project

Figure 2 depicts a development office, where engineers develop modules, hardware descriptor tables, and units made from modules. Figure 3 refers to the process in which new manufacture machine controller software can be automatically generated.

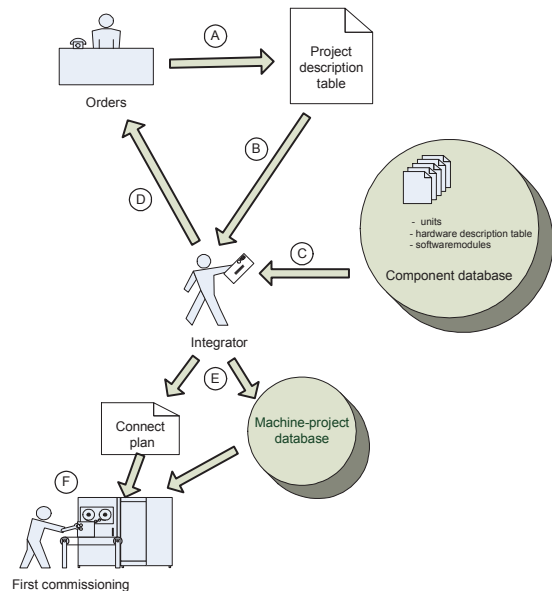


Figure 3. Simplified workflow of a project realization

Following the order A the customer specific production machine description is present. B the technician doing the integration checks the components repository C for the required components in an automated way. Then a software tool generates the

project database and the connected plan **E** for the machine controller. These passed to the installer staff **F**. After the integration there can be some feedback **D**.

Note, that the suggested development process implements the modules only once. These form a module repository so that the aforementioned advantages can be achieved.

When a new module is required to be developed the first step is to analyze the functionality. Experiments show that projects including similar modules have the same level of similarity. If the similar functions are implemented separately then it leads to redundant work and more opportunity of errors.

Engineers would need a tool which finds similar existing modules which implement the required functionality. The use of Group Technology can accomplish this to replace the human expertise.

As manufacturing machines have been created for many years now, Group Technology can be suitable for analyzing existing project. Very likely a project analysis including module similarity check will exhibit high level of redundancy.

In the following chapter the computer tools and mathematical models supporting these analyses will be detailed.

#### IV. AUTOMATED CLUSTERING METHODS

Automated clustering methods should be computer algorithms with the capability of clustering the controller code of the machines. There can be two different approaches. The top-down approach uses the project description table and extracts the module information. Then it codes the functional components and tries to find similarity in the module/unit repository. This approach can be used during the tendering phase of a project, when the actual machine and its controller do not exist. The other approach is the bottom-up approach where the controller code should be investigated. PLC coding can be done in various ways; some uses icons like Relay Ladder Diagram (RLD). Some code has textual format like the Structured Text (ST) form. This is suitable for GT clustering.

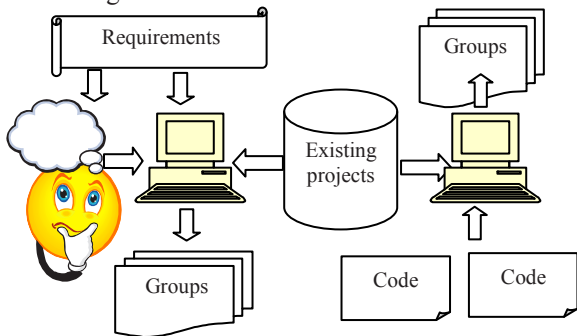


Figure 4. Top-down and bottom-up approaches

##### A. The bottom-up method

Structured Text (ST) is a PASCAL-based high-level programming language, based on the IEC 61131-3 standard, which standardizes programming languages for programmable controllers (PLC) [11].

ST is based on the Structured Text part of this standard. Using a high-level language like ST to program control systems offers the user a wide range of possibilities, for example:

- data management,
- process optimization,
- mathematical/statistical calculations.

In addition to IEC 61131-3 compliance, the some ST programming language also contains commands for manufacturing controller devices, motion control and technology.

Technology objects represent a technological functionality, e.g. positioning an axis or assigning parameters for an output cam. Technology commands are language commands provided by the technology objects. Such commands may be used, for example, to activate camming or to control motion sequences, for example, in order to position an axis.

The execution of user programs can be time-driven if you want them to run synchronously with the system clock or a defined time cycle. They can be interrupt-driven if they are to start and run once in response to a particular event. Alternatively, they can run sequentially or cyclically at the round robin execution level.

As the manufacturing machines are usually produced in small batches, their manufacturers make big efforts meeting the customer's specific needs and the modularization of the manufacturing machines

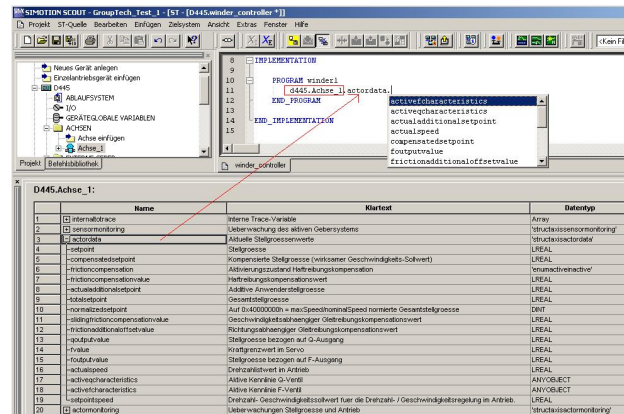


Figure 5. Meta data of the Structured Text based approach

According to the experience of the authors there are two widespread concepts:

- Standard Library Concept: the machine descriptor and controller application contains all the related controller software module of the manufacturer. On the installation and initial set-up of the machine these are parameterized to the specific hardware configuration by services provided by Structured Text (precompiler).
- Modular Software Concept: during the project database generation of the manufacturing machine only the selected software modules are transferred which are required to control the hardware.

Figure 5 shows the project screen of a ST based controller code. The highlighted piece of information is meta data, which can be extracted. ST based grouping of the project can be based on this.

**B. Mathematical formulization of the top-down problem**

Lets assume a binary polycode coding system. Each bit of the k bit long code represents a feature. 0 means that the controller does not implement the feature, 1 means that the feature is realized. Let's investigate n projects. The problem can be represented by an n x k matrix M, see for example (1).

$$\begin{matrix}
 1 \\
 2 \\
 3 \\
 4 \\
 5 \\
 6 \\
 7 \\
 8 \\
 9 \\
 10
 \end{matrix}
 \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
 \end{bmatrix}
 \quad (1)$$

Let pi represent the i<sup>th</sup> row. We can define a distance function by which we can evaluate how similar is one project to the other. For the sake of simplicity we assigned the decimal representation of the pi vector. Let dec(pi) denote this assignment.

Note, that this calculation is easy to execute, however it may give wrong result. For example

$$\begin{aligned}
 p_1 &= 0111111111 \text{ is closer to} \\
 p_2 &= 1000000000 \text{ than to} \\
 p_3 &= 0111111001.
 \end{aligned}$$

So a clustering the distance function dec(pi) will not give us a correct result, however if we order the pi vectors according to their decimal representation then this can be a starting point of further clustering.

This method however will find easily if two vectors are exactly the same:  $p_i = p_j$  if  $\text{dec}(p_i) = \text{dec}(p_j)$ .

**C. Genetic algorithm clustering**

The rows in matrix M need to be rearranged so that there will be block of 1s near in the p distance of the diagonal item. [4] details two measures of the goodness of the grouping, called Grouping Efficiency and Grouping Efficacy.

Grouping Efficiency is calculated as

$$\eta = q \cdot \eta_1 + (1 - q) \cdot \eta_2 \quad (2)$$

where

$\eta_1$  is the proportion of 1s in the diagonal block,  
 $\eta_2$  is the proportion of 0s in the off-diagonal block,  
 $q$  is a weight coefficient.

Grouping Efficacy is calculated as

$$\mu = \frac{N_1 - N_1^{out}}{N_1 + N_0^{in}} \quad (3)$$

where

$N_1$  is the total count of 1s in the matrix,  
 $N_1^{out}$  is the count of 1s outside the diagonal blocks,  
 $N_0^{in}$  is the count of 0s inside the diagonal blocks.

A Genetic Algorithm Clustering (GAC) was implemented in the form of MATLAB code. Both the Grouping Efficiency and Grouping Efficacy were implemented as a fitness function.

The n number of rows and k number of columns had been split to y and x number of block  $w_1, w_2, \dots, w_y$  and  $u_1, u_2, \dots, u_x$  so that

$$\sum_1^y w_i = n \quad \text{and} \quad \sum_1^x u_j = k \quad (4)$$

If the GAC would create an instance which does not meet (4) then the fitness function will be 0. Otherwise the fitness function is calculated as described above.

The pseudo code of the GAC is as follows

```

Create the initial set of the population.
Evaluate each of them
While not to stop do
    Select two random elements
    Crossover random genes of those two
    Mutate the new specimen randomly
    Evaluate the new specimen
    The new population will be the top elite
    of the previous population plus the bests
    of the new specimens
End while
    
```

After the block diagonalization process the (1) matrix should look as in (5).

$$\begin{matrix}
 8 \\
 6 \\
 3 \\
 2 \\
 4 \\
 1 \\
 9 \\
 7 \\
 5 \\
 10
 \end{matrix}
 \begin{bmatrix}
 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
 \end{bmatrix}
 \quad (5)$$

You can see three groups identified by the algorithm here:

- Group 1 consists of (8, 6, 3, 4),
- Group 2 consists of (1, 9, 7),
- Group 3 consists of (5, 10).

**CONCLUSION**

Group Technology is a well known concept in cellular manufacturing. The advantages of the GT can be applied in other areas like PLC code development. This paper presented two approaches for grouping. The Genetic

Algorithm Clustering prototype has been implemented and shows promising results for the further development.

#### ACKNOWLEDGMENT

The research and development summarized in this paper has been carried out by the Production Information Engineering and Research Team (PIERT) established at the Department of Information Engineering of University of Miskolc, Hungary. The authors are grateful for all the help they got.

#### REFERENCES

- [1] James T. L., Brown E. C., Keeling K. B.: A hybrid grouping genetic algorithm for the cell formation problem. *Computers & Operations Research*, 2007, 2059-2079.
- [2] Cheng C. H.: Algorithms for Grouping Machine Groups in Group Technology. *Omega*, 20(4), 1992, 493-501.
- [3] De Lit P., Falkenauer E., Delchambre A.: Grouping genetic algorithms: an efficient method to solve the cell formation problem. *Mathematics and Computers in Simulation*, 51, 2000, 257-271.
- [4] Gonçalves J. F., Resende M. G. C.: A hybrid genetic algorithm for manufacturing cell formation. AT&T Labs Research Technical Report TD-5FE6RN; 2002. 1-30
- [5] Kamrani A. K., Logendram R. (editors): *Group Technology and Cellular Manufacturing, Methodologies and Applications*. Gordon and Breach Science Publishers, 1998.
- [6] Suresh N. C., Kay J. M. (editors), *Group Technology and Cellular Manufacturing, State-of-the-Art Synthesis of Research and Practice*, Kluwer Academic Publishers, Boston-Dordrecht-London, 1998
- [7] Kusiak A.: *Intelligent Design and Manufacturing*. John Wiley & Sons, New York-Chichester-Brisbane-Toronto-Singapore, 1992.
- [8] Wang J, Kusiak A: *Computational intelligence in manufacturing handbook*. CRC Press LLC, 2001.
- [9] Hornyák O., Sáfrány G: A csoporttechnológia alkalmazásának vizsgálata moduláris gyártógépek vezérlőkódjának automatikus generálására, *MicroCAD'2009 International Scientific Conference*, Miskolc, pp. 2009, 43-49,
- [10] Hornyák O., Barthol G.: *Soft Computing Methods for Behaviour Based Control*, *MicroCAD'2008 International Scientific Conference*, pp. 2008, 165-170
- [11] Siemens AG.: *SIMOTION SCOUT online help (V4.1.1.0)*, 2007, pp 845.

