

Reusability of Function-oriented and Object-oriented Master Control Software

J. Uhl, J. Driller

*Institute of Control Technology for Machine Tools
and Manufacturing Units (ISW)*

Seidenstr. 36, D-70174 Stuttgart, Germany;

Tel. +49 711/121-2420, Fax +49 711/121-2413;

Email: [joachim.uhl|juergen.driller]@isw.uni-stuttgart.de

Abstract

Master control systems of the 5th generation are characterized by a change in paradigm in manufacturing technology. This change in paradigm is adapted automation, decentralized or human-centered. The necessary application flexibility and the possibility of forming variants require a high reusability of master control software, as well as a systematic development process. This paper shows as an example the software techniques which have been selected to develop a decentralized structured object-oriented master control system. A comparison is made with function-oriented master control systems.

Keywords

Object-orientation, master control system, software technology, reusability

1 INTRODUCTION

Control software must be viewed as a machine element which is to run through a systematic design and production process. Methods, procedures and development tools are used in this software design and production process (Siewert et al., 1994). This is generally called Software Engineering. However what is understood by software technology is application-oriented software engineering which uses methods and procedures tailored to the application.

In the development of control software, a differentiation is made between the application of software techniques for PLC and for NC control software in the machine control level and for master control software in the master control and cell control level. This paper is concerned with software techniques in the master / cell control level.

Apart from the general requirement of creating control software with systematic, engineering procedures and methods, the development of new software structures for master control systems of the 5th generation (Storr, Uhl, 1995; Pritschow et al., 1995; Driller, 1995; Adiga, 1993; Nof, 1994; Veeramani et al., 1993) influences in particular the development and application of software techniques for the master / cell control level.

2 REQUIREMENTS MADE ON MASTER AND CELL CONTROL SYSTEMS AND THE REQUIREMENTS RESULTING FROM THESE MADE ON THE SOFTWARE TECHNOLOGY

Requirements made on the 5th generation of master and cell control systems are an **individual automation adapted for each application**, a high flexibility, a simple adaptation to changes in in-house order control and resources (in particular manufacturing equipment) and a high reusability with the option of forming variants for a favorable cost-profit ratio of the master control software (Storr, Uhl, 1995; Uhl, 1995; Driller, 1995). These requirements can be fulfilled by (Uhl, 1995; Driller, 1995):

- a new software structured for control systems which is essentially reflected in engineering "physical" equipment and
- a development of master system applications based on a software modular system.

Figure 1 compares the software structure of a conventional master control system of the 4th generation (Brantner, 1993; Siewert, 1994) with function-oriented program modules and a database separated from the program modules of the decentralized, object-oriented software structure of a master control system of the 5th generation.

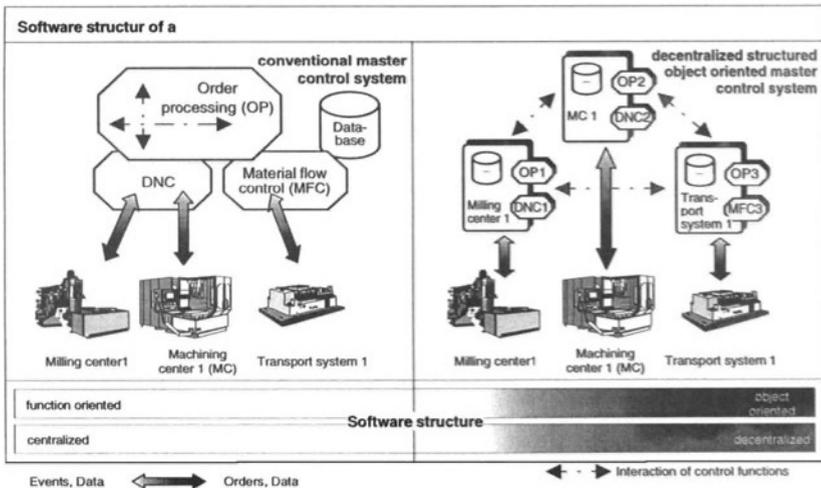


Figure 1 Comparison of software structures

A **decentralized software structure** must be differentiated conceptually from a **decentralization in machine technology** and a **decentralization in an in-house order control (organizational)** and refers to the division of coordinating master control functions such as order execution of software objects. Apart from coordinating master control functions, a master control system also possesses executive master control functions such as the DNC master control function.

While mostly traditional procedures (e.g. application of the waterfall model) and methods (e.g. structured analysis, entity-relationship, structured design, etc.) of software engineering were used for master control systems of the 4th generation, then in the master control systems of the 5th generation, methods and procedures adapted correspondingly are required in order to create time-effectively an inexpensive software.

The methods and procedures must guarantee (Storr, Uhl, 1995; Uhl, 1995; Driller, 1995):

- a structuring of the software as per flexible production system (FPS) components with a systematic, engineering decomposition of the FPS components and the representation of these relationships (see also Fig. 2),
- a structuring of the software according to master control functions,
- high consistency of the description methods used, i.e. a consistent application of information in all phases of the software engineering process,
- support of a simple reusability of the master control software from a software modular system,
- master control software which is easy to understand.

Figure 2 shows an example of the structure of the DNC master control function for a manufacturing system with two machines. The objects of the representation have been derived from a physical view of the manufacturing system to be automated.

A library is used as development platform to support the development work. The platform contains software for **communication** (between objects and also between processors), a **database** and **help functions** (e.g. list administration). The library will be used by all software developers for developing master control software. A commercially supported platform must be used. The main advantage of such a platform is the guarantee of computer independence arising from the encapsulation of operating system routines etc. This relieves software developers of routine programming work and he/she can then concentrate on the actual application. Another advantage is that the library routines will have been well tested, so software quality should improve.

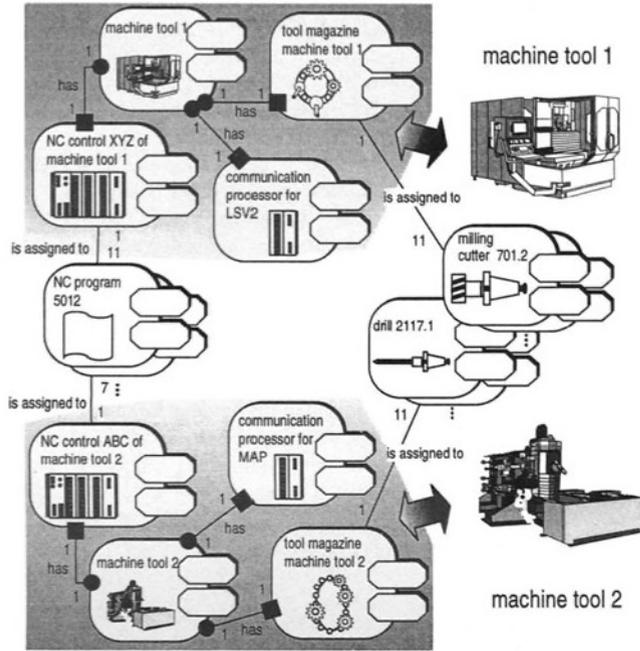


Figure 2 Example of the structure of the DNC master control function

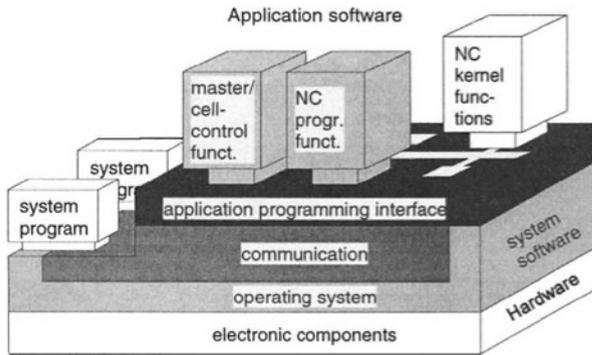


Figure 3 Application and structure of a platform

3 SOFTWARE TECHNOLOGY FOR MASTER CONTROL SYSTEMS OF THE 5TH GENERATION

3.1 Procedures for creating a master control system based on a modular software system

Development of a master control software using systematic design is required. A master control system of the 5th generation structured according to objects offers a good starting point due to its orientation towards "physical" equipment. This orientation facilitates the "design" of future master control systems for specific applications. The basis for this approach is the application of procedures and methods of object-oriented software engineering.

Figure 4 shows a procedure for developing an application-specific master control system in two stages. The development of the **master control software** is to be differentiated from the generation of the **master control system**. The procedure can be classified in the object-oriented software life cycle with the phases system analysis, system design, coding and module test as well as system test and use. The aim is the development of classes of the **class library of the master control software**. The classes of the **class library of the development platform** which are usually only available as source or binary code must already be taken into consideration in the module design and are included using reverse engineering.

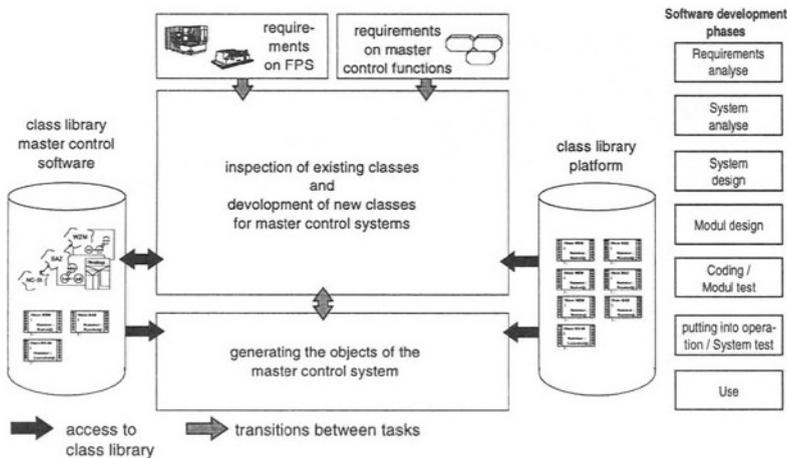


Figure 4 Procedure for creating a master control system

Figure 5 shows how the phases of the software engineering process will proceed. The **initial development** and the **follow-on development** of a master control system are differentiated as well as an **incomplete** and a **complete master control system class library** for generating the objects of a master control system.

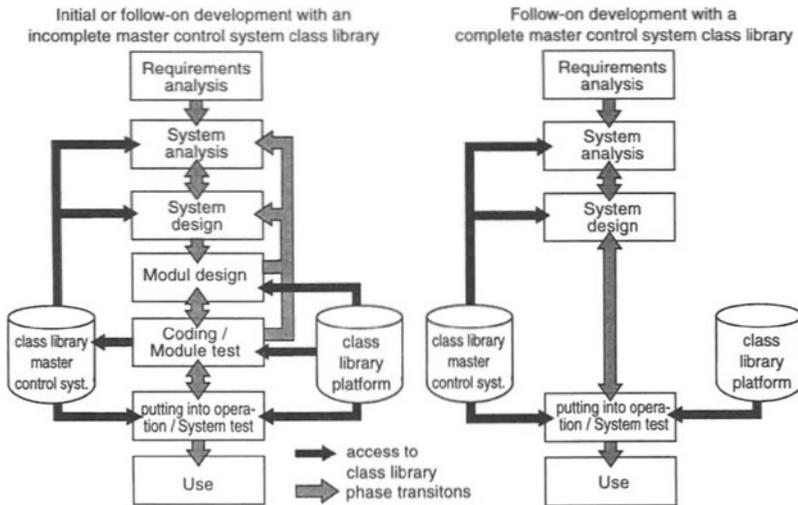


Figure 5 Differentiation of a development with an incomplete and a complete class library for the classes of the master control software

If the class library does not exist for the classes of the master control system in an initial development or is not complete in a follow-on development for a different application, all phases of the software engineering must be executed. A new development or adaptation of classes of the master system class library is necessary. If the control system can be created completely from the master system class library for a follow-on development for another application, then phases of the software engineering process as seen on the right of Fig. 5 can be skipped. Therefore in the system design phase, for example, the objects required and instantiated from the class library for the master control system only have to be checked and defined.

Figure 6 describes the systematic steps for developing a master control system. Note that the application-specific steps

- requirements made on FPS components and
 - requirements made on master control functions
- are the starting point. The transitions between the steps are progressive. The procedure includes frequent forward and backward movement between the steps.

The following are essential techniques in system analysis and system design:

- Decomposition of executive master control functions and FPS components
- Decentralization of coordinating master functions to FPS components.

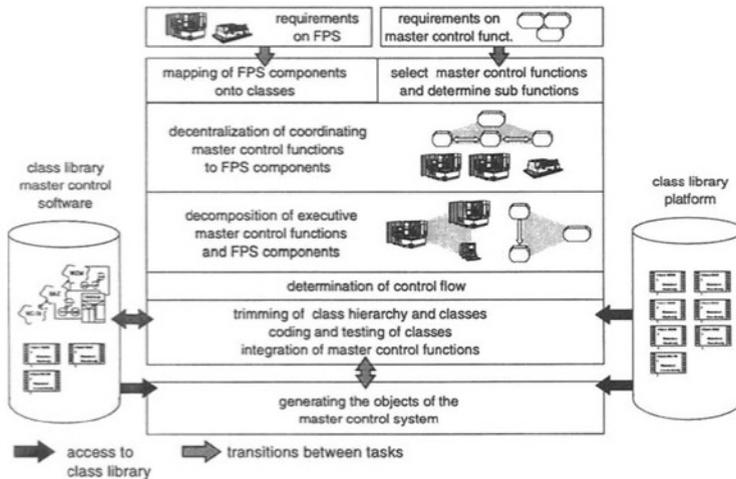


Figure 6 Systematic steps for developing classes for a master control software

3.2 Descriptive methods used

The descriptive methods that are used have a considerable influence on the consistency achieved between a system analysis, system design and coding. The development platform has an influence on the consistency between module design and coding. An important prerequisite for consistency is that an object-oriented descriptive method and an object-oriented development platform are used, i.e. that objects, classes, inheritance, aggregation and association are supported.

The **descriptive method of G. Booch** (Booch, 1991) has been used in the core phases, where actual software generation occurs, due to:

- its high consistency up to the option of code generation,
- a suitable notation (e.g. possibility of representing objects, classes, inheritance, aggregation and association relationships)
- its intelligibility etc.

In the following discussion it is assumed that the reader is familiar with the Booch notation. This however has been extended by **Nassi-Shneiderman diagrams** for the structured formulation of the methods of a class. Figure 7 shows the descriptive methods used in the various phases.

Figure 8 shows an example of the application of the descriptive methods for the decomposition of the FPS components of the machining center together with the DNC master control function from Figure 3. The decomposition is made corresponding to the decomposition during the **development of PLC software** in the machine control level. State diagrams are assigned to each class. Dynamic behavior, however, is not implemented as the switching between status, but by method calls on the objects in the object diagram, such as those in Figure 10.

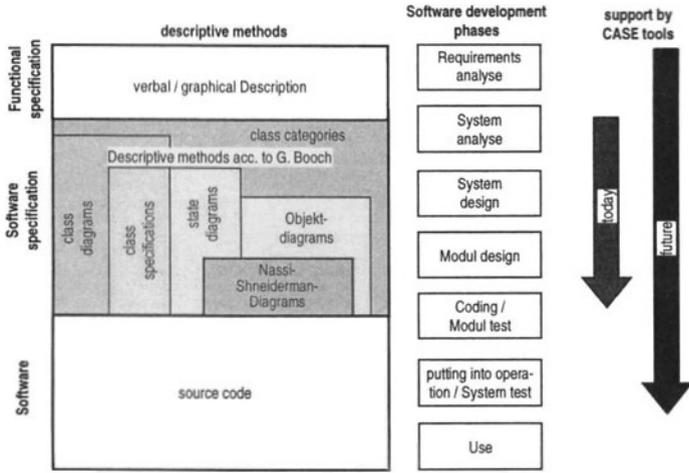


Figure 7 Descriptive methods used

Figure 9 and Figure 10 show examples of the use of a software tool for the descriptive method according to Booch. In Figure 9, the master control functions of a master control system are shown in class categories. The classes of the DNC class category are shown in Figure 10. The representation of a run for transferring an NC program to a machining center can be seen in Figure 11.

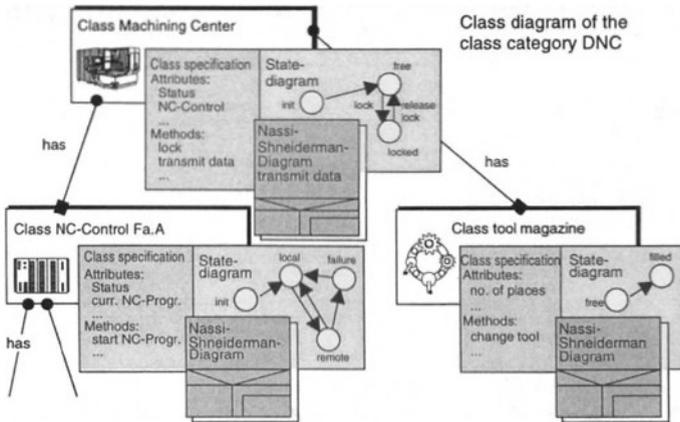


Figure 8 Engineering decomposition and application of descriptive methods for representing a machining center

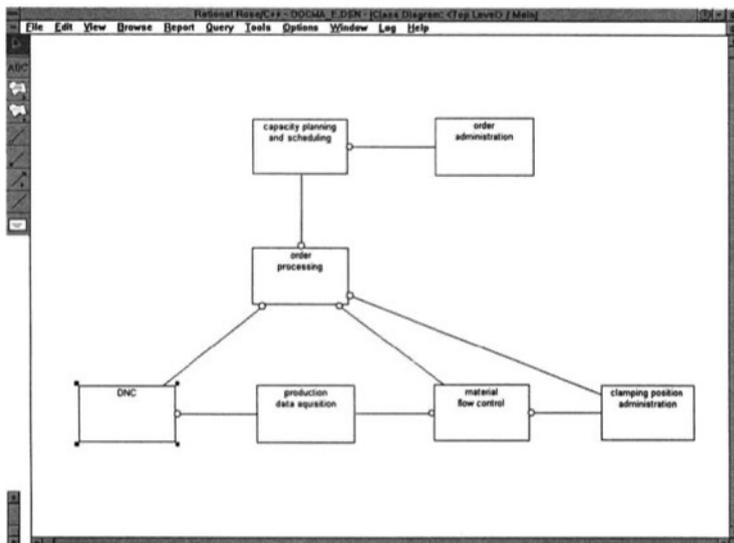


Figure 9 Class categories in a master control system

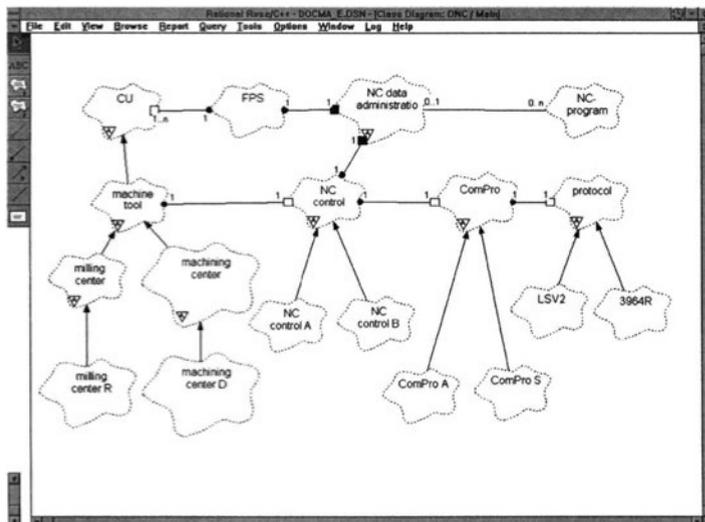


Figure 10 Example of a class diagram for DNC master control function

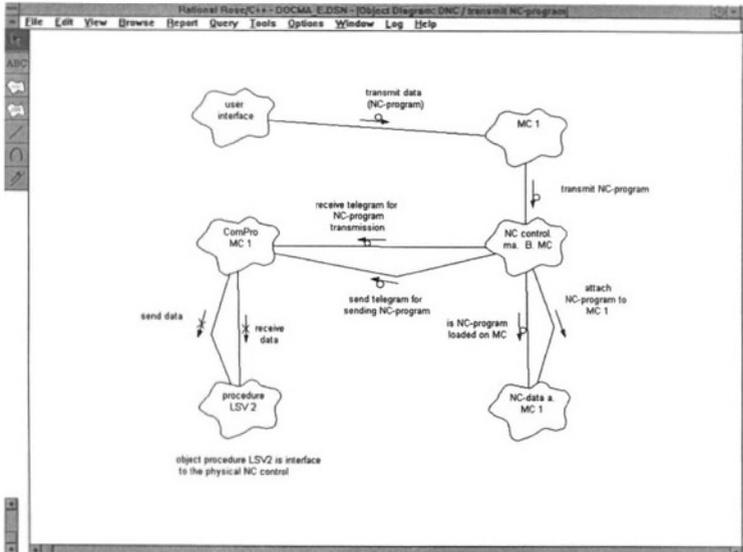


Figure 11 Object diagram for the execution of the transfer of an NC program

A standardized platform such as a platform based on the **CORBA standard** (Common Object Request Broker Architecture) or the **OSACA platform** (Open System Architecture for Controls within Automation Systems) should ultimately be used as the development platform. A development platform based on the CORBA standard has the advantage that it supports an object-orientation with objects, classes, inheritance, aggregation and association relationships across computers. An OSACA platform enables interoperability with objects at the machine control level.

4 IMPLEMENTATION EXAMPLE AND COMPARISON WITH FUNCTIONAL MASTER CONTROL SYSTEMS

The application of the software techniques discussed in this paper took place at ISW in the framework of prototype work on the decentralized structured, object-oriented production master control system for manufacturing applications (DOCMA). The CASE tool ROSE developed by Rational was used for modeling the master control software. The DSOM (Distributed System Object Model) platform developed by IBM based on the CORBA standard was used as the development platform. Figure 12 shows an implementation example.

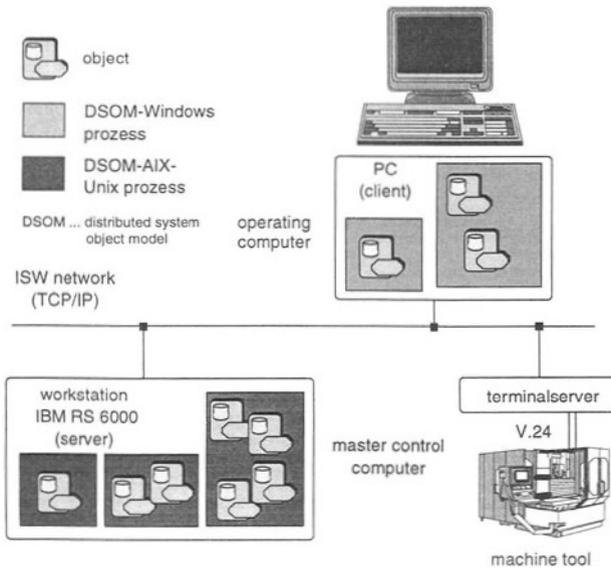


Figure 12 Implementation example

Table 1 shows a comparison of the decentralized object-oriented master control system DOCMA with the adaptable master control system ALSYS. ALSYS is a master control system of the 4th generation and was also developed at the institute. The comparison is for the DNC master control function for an FPS with 2 different machines and 1 transport device. It addresses the software techniques used and the reusability of master control software.

Criteria	ALSY5	DOCMA
Number of the phase transitions with a change of the descriptive method	Requirement analysis to system analysis, system analysis to system design, system design to coding	Requirement analysis to system analysis, system design to coding
Descriptive methods in system analysis:	Structured Analysis, Entity Relationship	Descriptive method acc. to G. Booch, Nassi-Shneiderman diagrams
System design:	Structured Design, Structograms	
Coding:	C	C++

Reusability of master control software	Function modules	Classes, inheritance
Reusability of master control software by	Configuration data, parametrization, editable sequence rules, etc.	Classes, inheritance, parametrization
Extension of master control software by master control functions	New function modules, extension of function modules	New classes, inheritance
Extension of the master control software by FPS components	Extension of existing function modules	New classes, inheritance
Archiving	Versions of function modules various FPS applications	Class library
average number of function modules / classes \ objects of DNC	approx. 3 - 5	approx. 14 - 18 \ 10 - 12
average reusability of function modules / classes of DNC within new applications	approx. 0 - 5	approx. 10 - 18
average number of source code lines per function module / class	approx. 14 000 - 55 000	approx. 100 - 2000

Table 1 Comparison of software technology and reusability for master control systems of the 4th and 5th generation

The key observations from the use of object-oriented software techniques in DOCMA also compared to functional software techniques in ALSYS can be comprised in the following points:

- The master control system is developed using a consistent software model and descriptive form. The software module does not need to be transformed during the transition from system analysis to system design.
- The creation of a library with classes for master control system applications is achievable. Object-orientation (abstract data types, inheritance mechanisms, aggregation etc.) offers an appropriate mechanism for this. However the consistent use of the descriptive methods is necessary in order to produce detailed documentation. Aids for managing the increasing number of similar classes must also be provided (e.g. good search mechanisms).
- Software changes to facilitate reuse of function modules for new applications cannot be avoided. Despite a high adaptability in functional master control systems, function modules are not universal. An object-oriented master control system on the other hand achieves a

high reusability due to a high number of classes with lesser functionality which can be easily exchanged.

- The descriptive methods in all phases must be based on the same object-oriented model (same mechanisms, e.g. inheritance, etc.) otherwise a transformation of the software model is necessary. This relates in particular to the transition from module design to coding. Examples of such transformations arise when we use
 - C instead of C++ (inheritance mechanisms have to be reproduced exactly),
 - a platform with interprocess communication instead of a platform with distributed object-oriented communication (object communication has to be represented via interprocess communication) or
 - a relational database instead of an object-oriented database (data and methods have to be separated for persistent storage in the application).

5 SUMMARY

This paper has given an insight into the use of object-oriented software techniques and systematic design in the development of a 5th generation master control system. It has shown how a reusable class library can be created for decentralized object-oriented master control systems and how application-specific master control system can be generated from the class library. The advantages of "designing" an application-specific master control systems with an interdisciplinary, physical view of the master control system have become clear. A comparison with functional master control systems was made and a simpler and higher reusability of classes compared to function modules was observed.

For the development of decentralized, object-oriented master control systems, consistent support by CASE tools is not yet available. Development work is particularly needed in the setting-up phase. This concerns the development of test environments with whose help the production system to be controlled is reproduced and simulated in order to check the operability of a master control system.

6 REFERENCES

- Adiga, S. (1993) (Ed.) Object oriented Software for Manufacturing Systems. London etc.: Chapman & Hall
- Booch, G. (1991) Object-oriented Design with Applications. Redwood City: Benjamin/Cummings
- Brantner, K. (1993) Adaptierbares Leitsteuerungssystem für flexible Produktionssysteme. *Dissertation Universität Stuttgart (ISW 96)*. Berlin, Heidelberg, New York: Springer
- Driller, J. (1995) Modellierung verteilter Steuerungs- und Informationssysteme in der Produktionstechnik. In: *Graduiertenkolleg Parallele und Verteilte Systeme, Zusammenfassender Bericht 1993-1994*, Stuttgart
- Driller, J. (1995) Föderative Steuerungssysteme in der Produktionstechnik. 3. *Berichtskolloquium des GK PVS*, Stuttgart

- Nof, S.Y. (1994) Critiquing the potential of object orientation in manufacturing. *Int. Journal of Integrated Manufacturing*, Vol. 7, No. 1, 3-16
- Pritschow, G.; Storr, A.; Handel, D.; Rommel, B.; Uhl, J. (1995) Uniform Object-Oriented Machining Model as Basis for Decentralized Planning in Master Control Systems. In: *Annals of the German Academic Society for Production Engineering*
- Siewert, U.; Reichenbächer, J.; Uhl, J. (1994) Softwarewerkzeuge für die Steuerungstechnik. In: *Fertigungstechnisches Kolloquium (FTK) '94*. Berlin, Heidelberg, New York: Springer
- Siewert, U. (1994) Systematische Erstellung adaptierbarer Leitsteuerungssoftware am Beispiel der Durchsetzungsplanung. *Dissertation Universität Stuttgart (ISW 100)*. Berlin, Heidelberg, New York: Springer
- Storr, A.; Uhl, J. (1995) Objektorientierte Leittechnik: neue Perspektiven und Lösungen. *CIM-Management*, 11, 30 - 34.
- Storr, A.; Reibetanz, T.; Uhl, J. (1995) Strukturierung von Zellenleit- und NC-Programmierungsfunktionen als Beitrag zu offenen Steuerungssystemen. In: *Offene Steuerungen*, München, Wien: Hanser
- Uhl, J. (1995) Leittechnik von morgen - Was bringt die Zukunft? In: *Effiziente, prozeßnahe Fertigungssteuerung und Überwachung*. ISW-Eigenverlag
- Uhl, J. (to be published) Beitrag zur Entwurfssystematik eines dezentral strukturierten, objektorientierten Leitsystems am Beispiel der DNC und Auftragsdurchsetzung. *Dissertation Universität Stuttgart*
- Veeramani, D.; Bhargava, B.; Barash, M.M. (1993) Information system architecture for heterarchical control of large FMSs. *Computer Integrated Manufacturing Systems*, Vol. 6, No. 2, 76-92

7 BIOGRAPHY

Joachim Uhl holds a master's degree in mechanical engineering. Since 1990, he has been working as research assistant at the Institute of control technology (ISW) at the University of Stuttgart and is head of the group 'production control systems and quality assurance'. His main research interests are the application of object-oriented techniques in manufacturing, the development of decentralized control systems and the development of open system architectures for cell controls.

Jürgen Driller holds a master's degree in computer science. Since 1993, he has been working as a research assistant at the Institute of control technology (ISW) at the University of Stuttgart. He is also a member of the *Graduiertenkolleg* 'Parallel and Distributed systems'. His main research interests are the application of object-oriented techniques in manufacturing and the development of decentralized control systems.