DATABASE OF MACHINING PROCESS PLANS IN CLIENT/SERVER ARCHITECTURE

Edward Gawlik, Jacek Habel

Edward Gawlik Ph.D., Jacek Habel M.Sc.
Cracow University of Technology, Institute of Production Engineering and Automation,
Al. Jana Pawla II 37; 31-864 Cracow, Poland
Tel. 0 48 12/ 648-05-55 ext. 3266, fax 0 48 12/648-20-10, e-mail: gawlik@mech.pk.edu.pl
Tel. 0 48 12/ 648-05-55 ext. 3239, fax 0 48 12/648-20-10, e-mail: habel@m6.mech.pk.edu.pl

ABSTRACT: *Conception of database of machining process plans in client/server architecture is presented in this paper. Characteristic of production system as well as its structure is included. General description of machining process plan as well as its structure, which was used to build database of machining process plans in client/server architecture, is also presented.*

1 INTRODUCTION

The designing of machining process plan is a set of activities, which belongs to preparing production system, it is one of the basic elements of product production system. It is connected with necessity to work out a structure of product production system, a general diagram is shown on figure 1. Basic elements (sub-systems) of this system are:

- Preparing product production system,
- Manufacturing product system.

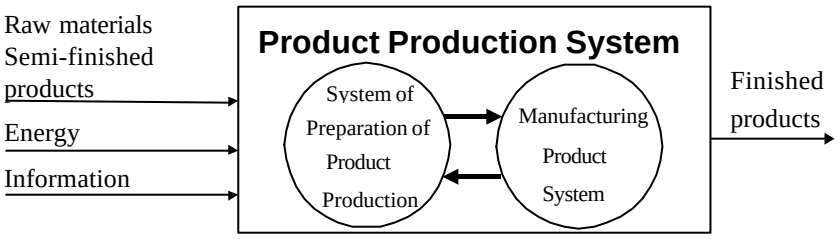

Figure 1. Diagram of product production system.

Raw materials, semi-finished products, energy and information are on the system input, finished products are on the system output.

2 STRUCTURE OF PRODUCT PRODUCTION SYSTEM

Structure of product production system has hierarchic character and could be represented in the form of a tree structure. On specified levels elements of product production system would be placed, which are connected between each other by branches. Description of this structure is as follow:

- *Zero level* – tree root, there is a production system,
- *First level* – on this level system elements are:
  - System of preparation of product production,
  - Manufacturing product system.
- *Second level* – on this level system elements are:
  - Elements connected with system of preparation of product production are as follows:
    - Product construction design subsystem,
    - Product machining process planning subsystem,
    - Product production process planning subsystem (scheduling subsystem).

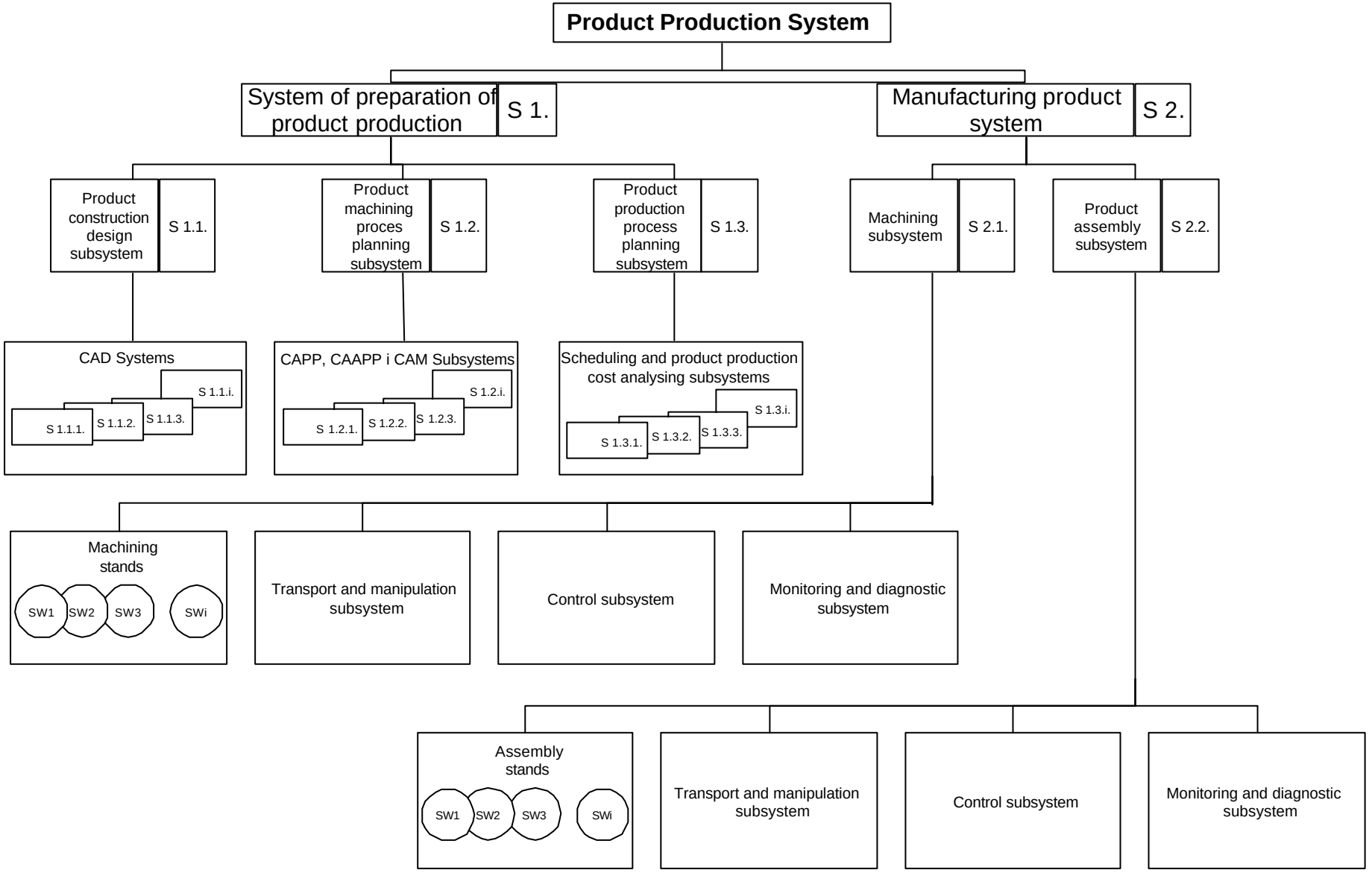

Figure 2. Structure of product production system

? ? Elements connected with product manufacturing system are as follows:
?? Machining subsystem,
?? Product assembly subsystem.

✍✍*Third level* – on this level system elements are:
? ? Elements connected with product construction design subsystem in the form of CAD (Computer Aided Design) system offered by different firms in the market,
? ? Elements connected with product machining process planning subsystem such as:
?? Machining process planning subsystem – CAPP (Computer Aided Process Planning),
?? Product assembly process planning subsystem – CAAPP (Computer Aided Assembly Process Planning),
?? NC programs generating subsystem – CAM (Computer Aided Manufacturing).
? ? Elements connected with product production process planning subsystem such as:
?? Product production scheduling subsystem,
?? Product production cost analysing subsystem.
? ? Elements connected with machining subsystem such as:
?? Machining stands,
?? Transport and manipulation subsystem,
?? Control subsystem,
?? Monitoring and diagnostic subsystem.
? ? Elements connected with product assembly subsystem such as:
?? Assembly stands for given assembly units as well as product,
?? Transport and manipulation subsystem,
?? Control subsystem,
?? Monitoring and diagnostic subsystem.

3 STRUCTURE OF MACHINING PROCESS PLAN

In product production system there is a realised product production process, whereas in product manufacturing system there is a realised product machining process. Product machining process plan includes [8]:

?? Set of process plans of semi-finished products for parts, which consists in product (apply only to semi-finished products, which are made by producer itself),
?? Set of machining process plans of parts, which consists in product (apply only to parts, which are made by producer itself) – realised by machining subsystem,
?? Product assembly process plan – realised by product assembly process subsystem.

Structure of machining process plan for a given workpiece, which consists in product, has a hierarchical character, similarly to a product production system structure, and can also be presented in the form of a tree structure. On specified levels of this structure elements of machining process plan (objects) would be placed, which are connected between each other by branches (figure 3) [7]. Description of this structure is as follow:

✍✍*Zero level* – root tree, there is a machining process plan of i-th workpiece.
✍✍*First level* – operations (machining, diagnostic and measurement, heat treatment etc.) are machining process elements,
✍✍*Second level* – there are set-ups of workpiece as elements of operation,
✍✍*Third level* – there are positions of workpiece as elements of specified set-up,
✍✍*Fourth level* – there are cuts as elements of set-up and position in given operation (especially for machining cuts),
✍✍*Fifth level* – there are passes as elements of specified machining cut.

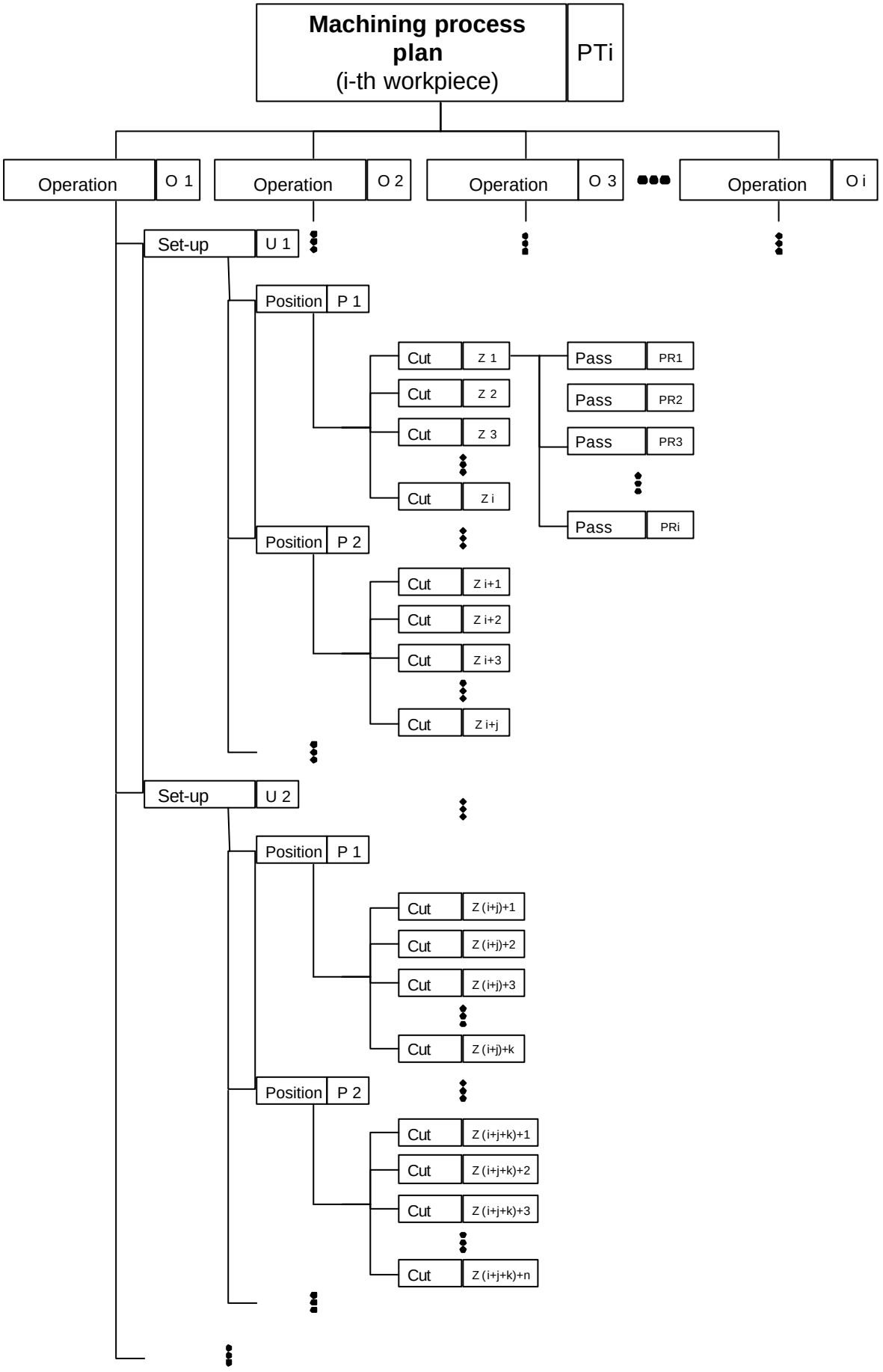

Figure 3. Structure of machining process plan

The structure, shown on figure 3, describes construction of the machining process plan and was applied as its model. The main advantage of tree structure is its ergonomics. Moreover it is preferred to apply object-programming technique.

4 Construction of database of machining process plans

Describing model of machining process plan we can easily affirm that it contain elements, which we can consider as objects (in the point of computer programming view) that possess some set of attributes (specific information about given process are stored). Depending on detailed level of saved information, these objects were presented in the form of hierarchical tree, where the root object (operation) has the most general information, and the lowest placed object (cut, pass) has the most detailed information (see figure 3).

Obviously every good worked out model should have its real representation. Because the main aim of this worked out model is application in computer aided planning, so it should have two representations [2, 4]:

?? Internal – object oriented, to do transformations in machining process plan during program run.

?? External – in the form of data base, to save process plan in computer disk memory.

Application of object databases is the best solution for representing machining process plan. It makes possible saving complex objects with its structure and remembering states of running methods (subprograms) associated with some attributes [3, 6]. But until now the standard of exchanging data through object databases wasn't worked out for technical reasons (especially conversion of execution code between different hardware platforms and different operating systems). We may use hybrid approach i.e. we may save a machining process plan using relational databases, which are freely available and easy convertible, and write a managing program using object oriented language [2, 4, 9]. As a result we obtain quasi-object database.

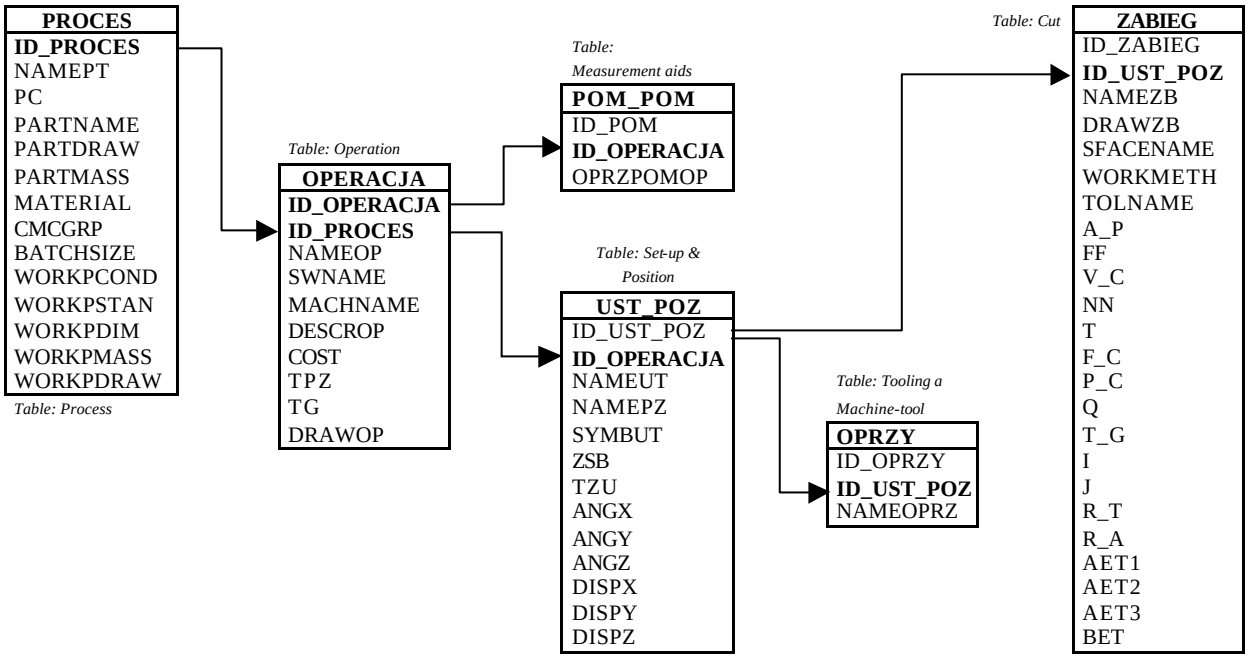

Figure 4. Machining process plan database scheme

Based on worked out model (figure 3), the experimental machining process plan database was developed, which general structure (scheme) is shown on figure 4.

The *Object Pascal* programming language is chosen for practical realisation of quasi-object database of machining process plans, which along with *Borland Delphi* environment is served to write database management system *DBMS*. DBMS is a set of procedures, which enable creating base and information processing included in it by the user. Defining base requires detailed

description of data, which will be saved in it, its kind, correctness control as well as coherence etc. Processing consist in searching and updating data and in preparing reports.

Another problem was connected with selecting concrete and the right kind database. Nowadays we can divide relational databases into two groups:

?? Desktop – designed for individual users [1]. Physical data as well as DBMS are installed on the same computer. Data are represented in the form of tables, which could be saved in one file (for example MS Access) or in many files (for example DBASE), where each file represents a separate table. Desktop databases do not give us mechanisms for outside access to information, that is the main limitation. If this database will be installed on several computers, then after some period of use we will have many different bases (includes different information). This is a big disadvantage.

?? Client/server – originally designed for accessing data in network environment [5]. Physical data as well as database logic are on the server, DBMS in the form of client application is installed on other computers connected to the local network. The main advantage of this approach is the possibility to use database by many users at the same time. Moreover in the client/server architecture there are implemented mechanisms which allow us to protect data by giving access rights on individual resources (tables, views etc.) and operations (writing, reading etc.) for all users and also mechanisms for data integrity protection (for example through using stored procedures).

For the sake of destination designed application (for example parallel work of some process engineers on different computers at the one project) the client/server architecture was chosen. Developed database consists of two, server and client modules.

The server module is a database application that runs on the computer, which fulfils physical server. The relational database model was chosen (figure 4) and was implemented in *InterBase* database server and the management program was written using *Borland Delphi 3.0* package. This module is designed to collect and store information entered through the client module. Moreover a manager module was created, which gives us the possibility to set users access rights on individual resources of this database.

The client module is an application that is designed to establish connection with server module and for database managing. Its module enables viewing into contents of process plans database, moving through process plans structure, edition and modification of individual records, adding and deleting object etc.

An example of screenshot of *ProcTech?* database, which contains structure of process plan number 10, workpiece characteristic and size of batch production is shown on figure 5. The window which includes basic characteristic of operation number 10 (for process number 10) is shown on figure 6. These characteristic includes:

?? Type of machine tool, where the operation 10 should be realised,
?? Used tooling a machine tool and workpiece fixturing,
?? Used measurement instrumentation,
?? Used tools,
?? Unit cost of operation etc.

5 Conclusions

The main advantages of *ProcTech?* program follows from differences between simple desktop databases and client/server. Differences are as follow:

?? Records in local table of database are physically ordered, however in client/server architecture we have oriented data set in specific structure,
?? Base division into tables in case of local base, in contrast to client/server, where base is hidden from users originally.

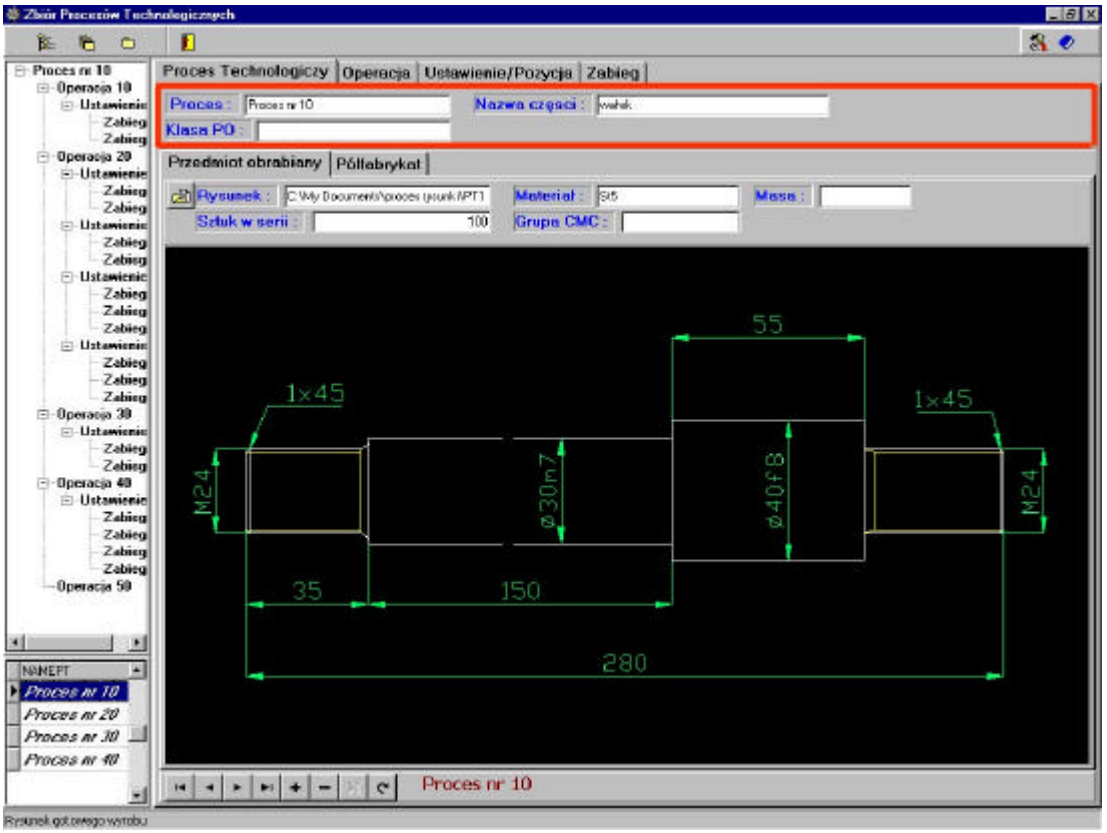

Figure 5. Screenshot of ProcTech? program which shows structure of process plan number 10 and drawing of finished workpiece

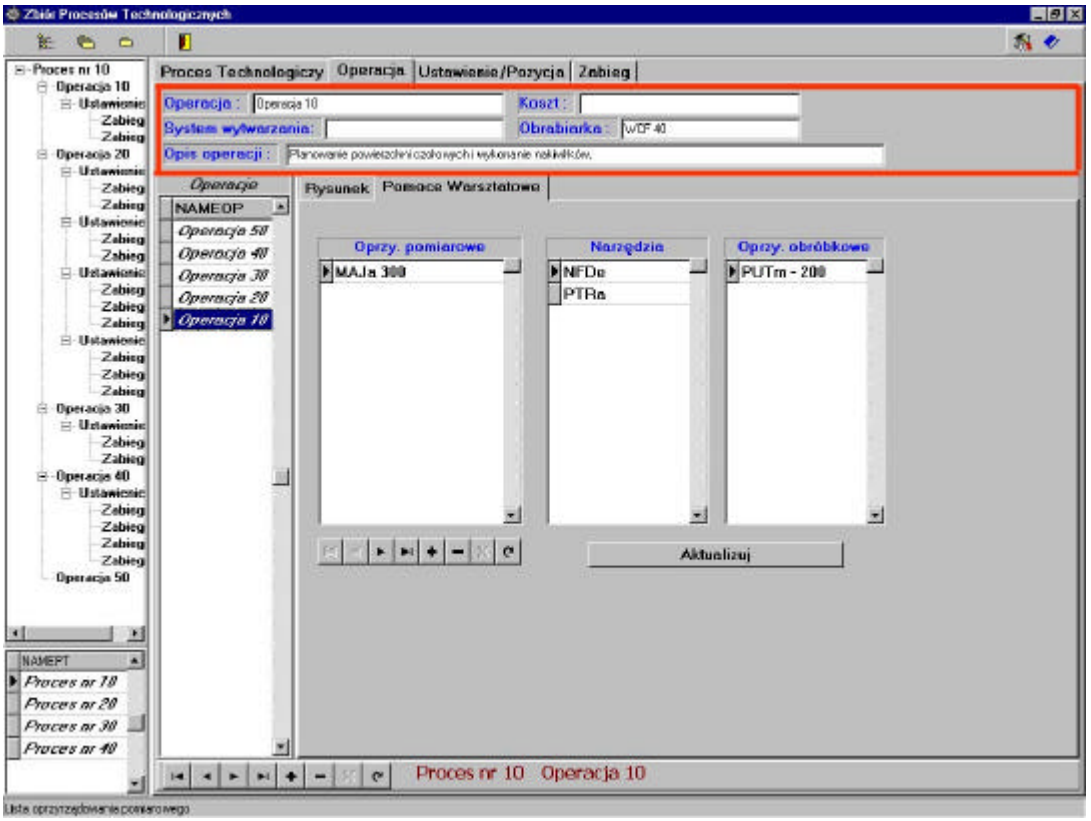

Figure 6. Screenshot of ProcTech? program which shows basic characteristic of operation number 10 in process 10

?? The client/server architecture data selection takes place in the server, which cause that only that portion of data goes through net to the client, which is really needed. It significantly take effects in reducing motion in the net comparing to situation when all databases are sent.
?? In case of local base exist possibility of limitation in database access entirely, it means on base (rule) – "all or nothing". There are no possibility to give preferential rights for users and groups of users. In client/server architecture this problem was solved very well.
?? Data integrity assurance in local databases depends on application that use base. In client/server architecture using data rules are defined on creating server stage. To this end the stored procedures are applied, which except execution of actions ordered by user, and execute additional actions required by integrity mechanisms.

6 LITERATURE

[1] Delobel C., Adiba M.: *Relacyjne bazy danych*. Wydawnictwa Naukowo-Techniczne, Warszawa 1989.
[2] Duda J., Habel J., Kwatera M., Samek A.: *Data Base with open architecture for defining Manufacture System Capabilities*. Flexible Automation & Intelligent Manufacturing FAIM'99, Tilburg, Holandia 23-25 June 1999.
[3] Figura D.: *Obiektowe bazy danych*. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1996.
[4] Habel J.: *Realizacja bazy danych mozliwosci technologicznych systemów wytwarzania dla potrzeb systemów CAPP*. XII Konferencja nt. „Metody i srodki projektowania wspomaganego komputerowo". Politechnika Warszawska, Instytut Podstaw Budowy Maszyn Warszawa, 24-26 listopad 1999.
[5] Henderson K.: *Bazy danych w architekturze klient/serwer*. Wydawnictwo Robomatic, Lódz 1998.
[6] Kim W.: *Wprowadzenie do obiektowych baz danych*. WNT 1996.
[7] Projekt Badawczy PB-254/T07/97/12 (kierownik J. Duda): *Opracowanie zalozen systemu projektujacego procesy technologiczne obróbki w warunkach inzynierii równoczesnej z wykorzystaniem sztucznej inteligencji*. ZADANIE 3: *Sformalizowany opis metod projektowania procesów technologicznych obróbki; wariantowej; semigeneracyjnej i generacyjnej*. ZADANIE 7: *Opracowanie zalozen budowy systemu projektowania procesów realizujacych projektowanie operatywne*.
[8] Samek A.: *Projektowanie procesów technologicznych obróbki skrawaniem i montazu*. Skrypt dla studentów wyzszych szkól technicznych. Politechnika Krakowska. Kraków 1986.
[9] Taylor D.A.: *Technika obiektowa*. Helion 1994.