Tool Management of the NX System

Summary

This article shows methods for creating applications for CAD/CAM systems. The program which fulfills the function of Tool Administration for the NX System was written in Java. The article also discusses basic ideas related to technological databases, as well as classification of cutting tools: milling, and drilling.

Keywords: technological databases, application, CAD/CAM systems

Introduction

In industrial production, the demand for higher productivity and better quality products is on the rise. In the manufacturing process, the selection of proper cutting tools and their parameters is crucial to the speed of production and quality of the resulting parts. Fast and optimal selection of the aforementioned factors and flexible usability of the tool management system are of great importance. Speedy and reliable flow of information, which affects the preparation time of production, is of great concern. Modern tools are equipped with information systems, which allow for their identification as well as gathering and handling data related to molding. Have appropriate software connected to a technological database one can quickly and efficiently manage the tools of the trade.

There is a lack of software that provides an easy way of entering tools into systems that aid in the molding process, such as the NX system. Currently, adding tools occurs externally to the program, which decreases its usability. A solution to this problem is presented in this article as an application which allows for an easy method of adding tools into the discussed CAD/CAM system. An application named "Technolog" was created as the tool management system of the NX system. The user of this application has instant access to all cutting tools as well as their full specifications.

The majority of this article is dedicated to introducing the "Technolog" application, which can be leveraged to manage cutting tools in CAD/CAM systems and can be applied to the industry.

1. Technological Database

The Technological Database, *TechDB,* is a collection of data that exists throughout a given period of time. This data is organized and stored within a structure that is defined by a database management system (DBMS). Information related to operations of a given firm or concerns of a given unit is also stored alongside this data.

Modern database technologies are a priceless tool to the industry. They allow for collecting, storing, and processing of information. The centralization of information decreases the possibility of mistakes and eliminates the need for redundancy. Databases primarily store useful information about machines and their corresponding tools, explanations about cutting methods and strategies, as well as specifications about speed and movement. Since technological knowledge is in one sense systematized across many categories (tools, machines, operations, materials), it requires a real, well-prepared technological database. Before selecting such a database, one must take into account several factors. One of these factors is adequate security system with sufficient access rights to satisfy a level of protection suitable for the

stored data. Technological databases must ensure safekeeping of data to prevent unauthorized access to it or malicious corruption of it.

Taking into account the number of different implementations of CAD/CAM systems throughout the industry as well as throughout universities and schools of technology, the importance of technological databases, such as Catia and NX, is very prominent. Such databases allow for prompt lookup of gathered information, modeling, computation and analyzation of 3-D models, programming molds and creating documentation based on carried out activities. Additionally, they allow for parameterized calculations while simultaneously comparing calculated results with standard values.

2. Tool Management in the NX System

The NX System is an advanced CAD/CAM/CAE system. It provides consistent and integrated tools for assisting work at each stage in the product development cycle. It begins with its initiating concept as a project, goes through verification (digital simulation), and finishes with production. It is developed on new technology that is based on the core system Parasolid. It is worth noting that NX's competing product Solid Edge is also based on this same core system.

NX is a solid and effective tool, setting the industry standards. As of today, the most recent version is NX6 and it is constantly being improved.

The CAM environment of NX consists of:

- a tool library
- a machine library
- a material library
- a library of rotational movements and speeds
- a set of post-processors
- workshop documentation templates
- part templates

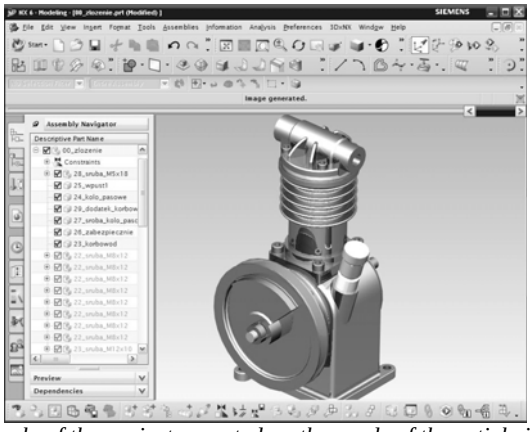


Fig. 1. The example of the project executed on the needs of the article in the system NX

Introduced on fig . 1 project was created stay from 29 elements which were specified in following composition (Fig. 2)

Fig. 2. The list of elements of the executed project in the system NX

The following part of this article discusses two possibilities of managing tools in the NX system, specifically those from the tool library. The first method deals with adding new tools to the program as well as the ability to export these tools into the technological tool database of the NX system. In order to add a new tool, one must select the "Create Tool" icon from the *Insert* toolbar. The available types of tools depend on the previously selected *"Template Parts"*. NX groups its milling and drilling tools in a proprietary way. The milling group of the NX system is presented in table 1, whereas the tools of the drilling group is presented in table 2.

Table 1. Milling tools, Drilling tools of the NX System

Icon	***Tool Name Milling tools***
	End Mills Non-indexable
	End Mills Indexable
	Ball Mills Non-indexable
	Face Mills indexable
	T-Slot Mills Non-indexable
	Barrel Mills
	UG-5 Parameter Cutter
	UG-7 Parameter Cutter
	UG-10 Parameter Cutter
	Thread Mills
	Mill Form Tool

Icon	***Tool Name Milling tools***
	Twist Drill
	Insert Drill
	Gun Drill
	Indexable Insert Drill
	Core Drill
	Spot Drill
	Center Drill
	Bore
	Counterbore
	Counter Sinking Tool
	Tap
	Chucking Reamer
	Taper Reamer
	Step Drill
	UG Drill

After selecting the appropriate tool one should enter its name in the *Name* field. Accepting the provided parameters using the "OK" button will cause one of several windows to open, where one can then enter information relating to the se-

lected position. The type of window that opens depends on the entered parameters as well as the selected drill or mill.

After entering all required fields, clicking on the "OK" button will accept the selected tool. To be able to reuse the same tool in more than one program, one must export it to the technological tool database of the NX system. One can perform this exporting by clicking on the icon located in the *"Library"* tab of the *"Create Tool"* window.

The second method for managing tools in the NX system deals with selecting the required tool from the tool database of the NX system. In the *"Create Tool"* window (fig. 3) one must click on the "Transfer from Database" icon, then clicking the "OK" button to confirm.

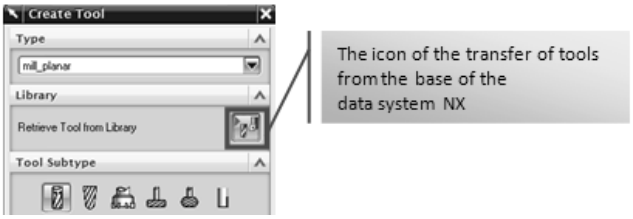


Fig. 3. Tool Selection from the NX tool database

Within the newly opened window *Library Class Selection*one can select a group of tools. Clicking on the "OK" button after entering all parameters will display a tool satisfying the input requirements.

In order to manually select a tool from all that are available, one should leave all fields blank and click on the "OK" button. Then, from the available list of tools, one should select the desired position. Double-clicking on the position's name will add it to the program.

The above-mentioned methods available in the NX software do not provide for sufficient management of tools. A problem arises when two or more organizational units need to use the same tool.

It is desirable to design a program that allows for easy and secure access (with the help of the latter described method) to tools tailored to specific needs of a user accessing the tool database.

3. Developing the Tool Application "Technolog"

Developing an application capable of managing the technological tool database of the System NX required locating and understanding the structure of the files that contain information about the various tools in the system. Trial and error, as well as personal experience, led in the direction of search for a single file that contains all the information about the tools used in the NX system.

The NX system stores data in files with *.dat extensions. Depending on the unit in which they were saved, information about tools is stored in various folders. The name are tied to metric customary units. Files with metric units are located in the folder:

…\UGS\NX6.0\MACH\resource\library\tool\metric

The full path depends on the installation folder. By default this folder is:

C:\Program Files\UGS\NX 6.0\MACH\resource\library\tool\metric

The tool database is stored in the file ***tool_database.dat***. A simple text-editor such as notepad is sufficient for editing the contents of this file. Adding new tools to this file or editing existing tools is relatively simple and straightforward, so long as the structure of the file is not altered. Each tool group is identifiable their relevant parameters. Columns #3 and #6 are key for this task. These columns define the relationship between a given group and a given tool.

Table 2. Sample file format of groups and subgroups of mills and drills in the NX system

T	Lib Type	ST	Lib Subtype	UGT	UG Type	UGST	Subtype
02	Milling	01	End Mills Non-indexable	01		01	5 Parameter
02	Milling	02	End Mills Indexable	01		01	5 Parameter
02	Milling	03	Ball Mills Non-indexable	01		04	Ball
…							
03	Drilling	01	Twist Drills	02		00	Standard
03	Drilling	03	Core Drills	02		00	Standard
03	Drilling	21	Spot Drills	02		04	Spot Drill

Using these columns the system is able to recognize and associate tools to their appropriate group. Each group of tools has its own subgroups, which are defined by a specific number. Tool data is entered into the file in a specific order. The system can recognize specific columns by the characters "|", which are used to separate the columns. Prior to the table of tool data are abbreviations and their definitions that are used in the tool data. The structure and format of the data block is very important.

```
# End Mill (indexable)
#
#  LIBRF    - Tool Library Reference
#  T        - Tool Type
#  ST       - Tool SubType
#  UGT      - UG Tool Type
#  UGST     - UG Tool SubType
#  DESCR    - Description
#  MATREF   - Tool Material Code
#  MATDES   - Tool Material Description
#  TLNUM    - Tool Number
#  ADJREG   - Tool Length Adjust Register
#  CUTCOMREG - Tool CUTCOM Register
#  HLD      - Tool Holding System
#  HLDDES   - Tool Holding System Description
#  DIA      - Tool Diameter
#  FN       - Tool Flutes Number
#  HEI      - Tool Length (Height)
#  ZOFF     - Tool Z offset
#  DROT     - Tool Direction (3=clockwise, 4=counterclockwise)
#  FLEN     - Tool Flute Length (Cutting Depth)
#  TAPA     - Tool Taper Angle
#  COR1     - Tool Corner1 Radius
#  HOFF     - Tool Holder Offset

#CLASS END_MILL_INDEXABLE
FORMAT LIBRF T ST UGT UGST DESCR MATREF MATDES TLNUM ADJREG CUTCOMREG HLD HLDDES DIA FN
# ----+-----------+--+--+---+---+-------------------+------+------+------+-------+---
DATA | ugt0202_001 | 02 | 02 | 01 | 01 | Insert Cutter 40 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_002 | 02 | 02 | 01 | 01 | Insert Cutter 25 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_003 | 02 | 02 | 01 | 01 | Insert Cutter 12 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_004 | 02 | 02 | 01 | 01 | Insert Cutter 25 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_005 | 02 | 02 | 01 | 01 | Insert Cutter 16 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_006 | 02 | 02 | 01 | 01 | Insert Cutter 12 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_007 | 02 | 02 | 01 | 01 | Insert Cutter 25 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_008 | 02 | 02 | 01 | 01 | Insert Cutter 25 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_009 | 02 | 02 | 01 | 01 | Insert Cutter 20 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_010 | 02 | 02 | 01 | 01 | Insert Cutter 20 mm | TMC0_00003 | P20 |    |
DATA | ugt0202_031 | 02 | 02 | 01 | 01 | Insert Cutter 50 mm | TMC0_00003 | P20 |    |
#END_DATA
```

Fig. 4. Screenshot of data from the NX database

Knowing the above information as well as the described structure for storing data in the data file, all that is left is deciding a way to enter and manipulate the data. Writing the "Technolog" application, a decision was made to use MySQL as the database backend for storing data.

All data is stored within the database. Except for specific data about the tools, the database stores all information about category names used in the side menu, as well as tool parameters.

The entire database contains 57 tables. The name of each table implies the nature of the data stored within the given table.

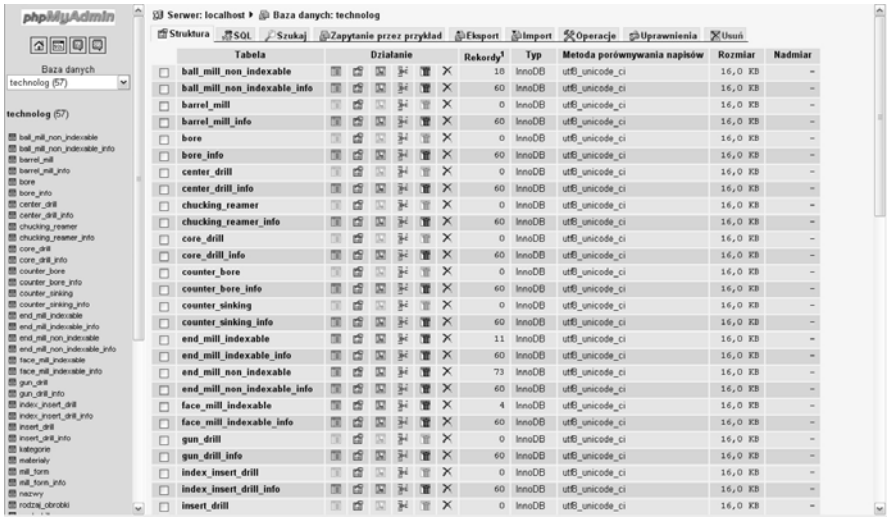


Fig. 5. Screenshot of the MySQL database used in the "Technolog" application

4. Tools used in developing the "Technolog" application

a) Java

Java is an object-oriented programming language developed by Sun Microsystems. The source code is compiled into byte-code, which is interpreted by the Java Virtual Machine (JVM). The Java programming language is similar to the C++ programming language, but Java does not allow for pointers. Furthermore, Java contains a "Garbage Collector", which automatically frees memory used by objects that are no longer needed by application.

A Java application can be initialized with the help of the *java* program, which initiates an instance of the JVM. The *java* program passes to the JVM the parameterized location of the byte-code – *java -jar program.jar* – or when the application consists of a set of compiled yet uncompressed classes – *java package.program.Main* – where *package.program.Main* is the main class of the application.

Often times it is necessary to reference additional libraries (such as JDBC drivers) which are usually enclosed in *.jar files. In such case, it is necessary to include the "-cp" option, which specifies the location of the additional files.

i.e. *java –cp lib/driver.jar package.program.Main*

In many modern systems where a certain file extension is associated with a specific program, it is possible to initialize a Java application by simply double-clicking on the icon of the program file. [1]

b) ECLIPSE IDE

Eclipse is an integrated programming environment for development Java applications. Thanks to many plugins that extend the functionality of the Eclipse IDE (Integrated Development Environment), it is possible to create software in languages such as Java, C++, PHP, Ruby, as well as others. Eclipse is based on the SWT library, which offers GUI (Graphical User Interface) components which interact with the operating system. Eclipse assists the programmer in developing software, and it is an Open Source project. The environment is free for all purposes. The current version of Eclipse is version 3.3 and it is codenamed "Europa" (the codenames of Eclipse version are the moons of Europe) [4].

c). JDBC API

JDBC (*Java Database Connectivity*) is a programming interface for communicating with databases from a Java application.

The Java platform (JRE) includes the JDBC API (*Application Programming Language*). In order for an application to be able to connect to a database, a database driver implementing an interface is necessary. Because of this design, connecting to a variety of different database is virtually the same process from the applications point of view. The only differences exist in the SQL commands sent to databases.

A different method for communication with database is the ODBC (*Open Database Connectivity*) interface. The API version of the JDBC driver specifies the version of the interface available in a given version a Java environment. API version 2.0 is available in JRE 1.1, API version 3.0 is available in JRE 1.4, and API version 4.0 is available in JRE 1.6 [2].

There are 4 different types of the JDBC driver:

1. Type 1 – JDBC-ODBC Bridge together with the ODBC library for a given database
2. Type 2 – a partial implementation of the JDBC driver in the Java language; the driver calls procedures from a library of a given database
3. Type 3 – a complete implementation of the driver in the Java language; however, this implementation communicates with a database through a middle layer, which can be used for several different databases
4. Type 4 – this is similar to the Type 3 driver, except this driver communicates directly with a specific database server, through protocols such as TCP

A connection to a given database is established by passing proper arguments to the *Drivermanager.getConnection()* method. The argument is standardized and looks like the following (example of a MySQL connection string):

jdbc:mysql:///[host][,failoverhost...][:port]/[database]
[?propertyName1][=propertyValue1][&propertyName2][=propertyValue2]...

The brackets [ ] represent optional parameters.
A specfic example:
jdbc:mysql:///localhost:3306/bazaDanych?user=root&password=admin

The connection string for JDBC-ODBC drivers looks somewhat different, for the reason that the interface is accessible only locally (for databases on the same machine):

*jdbc:odbc:<data-source-name>[;<attribute-name>=<attribute-value>]**

A specific example:

jdbc:odbc:mydb;UID=root;PWD=secret

Below is a short example of how to obtain a connection to a databse.

```
String className = “com.mysql.jdbc.Driver”;
String url = “jdbc:mysql://localhost:3306/database”
+ „?user=root&password=admin”;
String query = "SELECT * FROM " + tableName;
// load the driver
Class.forName(className).newInstance();
// open the connection
conn = DriverManager.getConnection(url);
stat = conn.createStatement();
// execute the query
ResultSet rs = stat.executeQuery(query);
// traversing all results
while (rs.next()) {
// perform some operation on the result, for example retrieve the result of column 1
String s = rs.getString(1);
}
```

JDBC also offers classes for discovering the structure of a given database using metadata. Since the driver ensures the implementation of these classes, acquiring metadata is performed similarly on all databases (with minor exceptions). Most metadata can be retrieved with the help of methods of two classes [2]:

- *java.sql.DatabaseMetaData* – information about a database, its configuration, tables, functions, users, and driver
- *java.sql.ResultSetMetaData* – information about a table returned by a query

5. Cutting tools on the example of drills and mills

An application for managing tools has been created. Included with it are tools that deal with cutting, including mills and drills. The regions categories have been synchronized with the tool management system of the NX system. The names of the categories standardized with firms that specialize in production of cutting tools.

The naming convention of cutting tools is not strictly regulated. Each producer uses their own conventions for different categories of products. The most widely used categories do match those of accepted standards. Until July of 2005 the names PN-ISO 11529-1 and PN-ISO 11529-2 defined mills. However, these names were removed without any opposition. Regarding current terminology, PN-93 M-01015 refers to mills and PN-93 M-01019 refers to drills. The above norms define the types, groups, installation, and construction of these tools. All cutting tools can be classified based on the following criteria:

- the cutting method
- the shaping method
- the direction of work
- the number of cutting tools
- the angle of mechanization
- the method of installation
- the type of the cut
- the primary movement
- the shape of the surface

- the type of construction
- the method of cooling
- the number of assignments
- the method of installing protective features
- the degree of dissemination
- the possibility of executive other moves

Since the number of possible constructions is very large and is continually growing, it can be classified based on many categories:

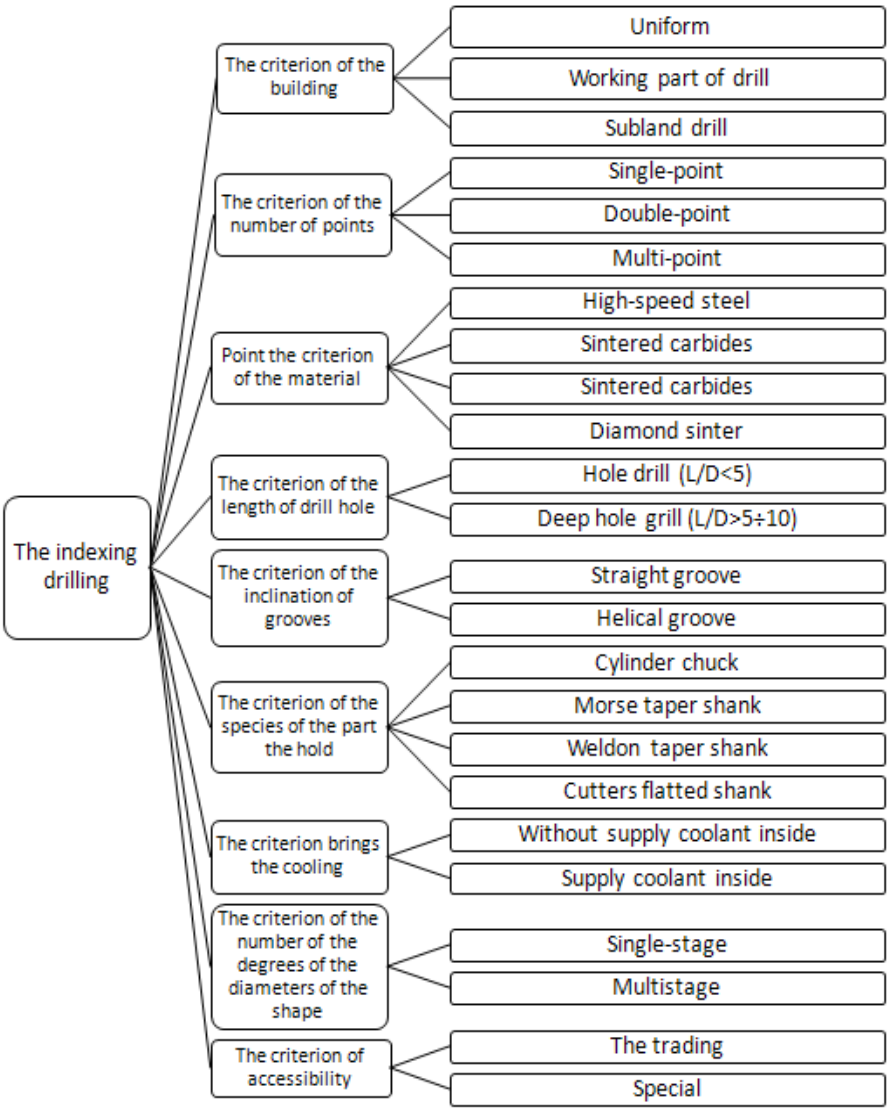
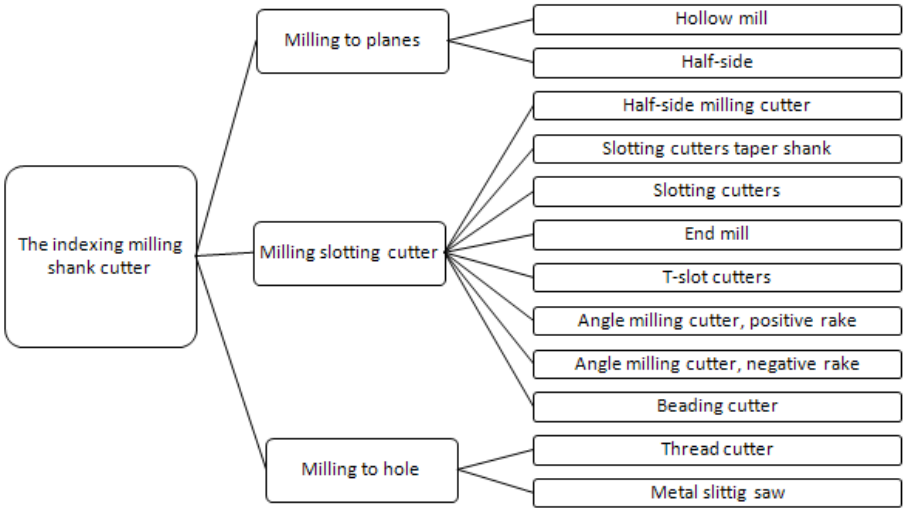


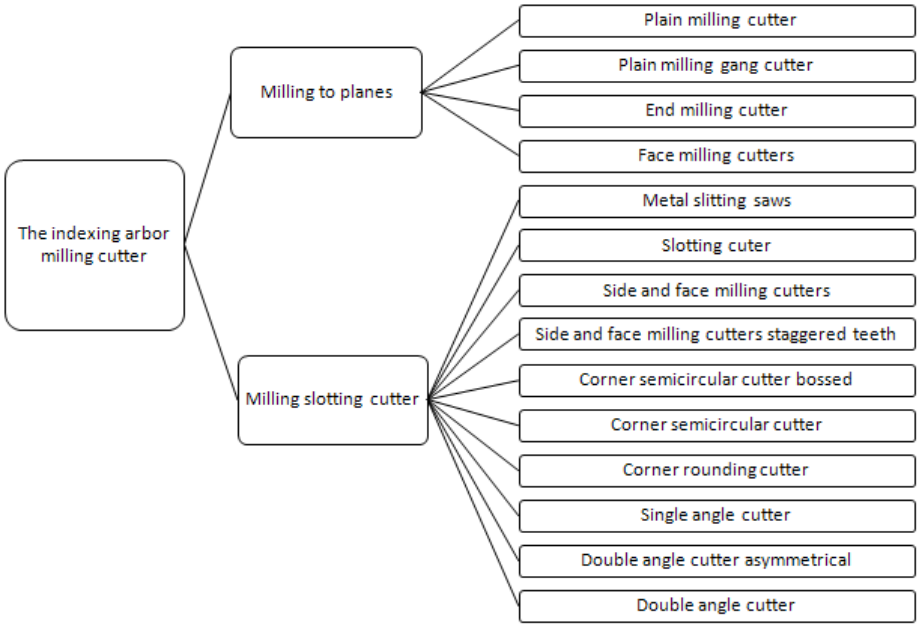
Fig. 6. Example classification of mills and drills [3]

Classification of mills according to PN-93/M-01015



Figs 7. Classification of molding mills according to PN-93/M-01015

Classification of plating mills according to PN-93/M-01015



Figs. 8. Classification of plating mills according to PN-93/M-01015

6. Working with the "Technolog" program.

This program was written in Java and it works together with the MySQL database. It runs on two servers, WAMPSERVR and Derby. An installer is included with the application to allow for an easy installation by the user.

Fig. 9. Starting installation window

Minimum program requirements:

- 400 MHz Procesor
- Microsoft Windows 2000/XP
- 256 MB RAM
- CD-ROM
- 18 MB of harddrive space
- Java driver

The program summary is presented based on the programs main window.

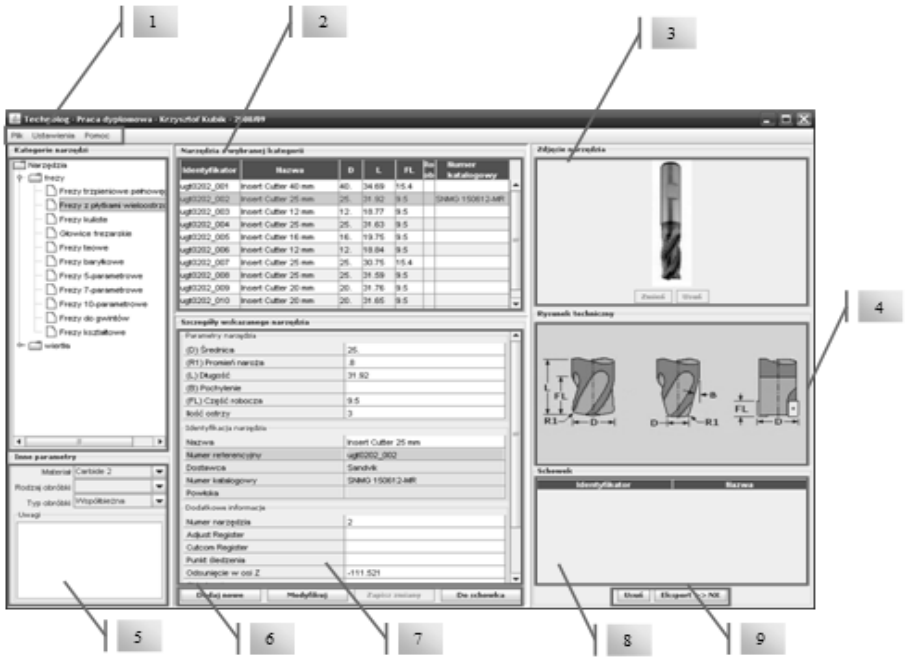


Fig.. 10. Main Program Window

Specific features:

1. ***Program menu*** consists of the following functions:
   - *File>> Import from XML* – allows for importing parameters from an XML file
   - *File>> Export NX* – exports stored parameters to NX-a

- *File>> Finish* – exit application
- *Settings>> Export Catalog* – allows setting up of a catalog file to which stored data should be exported
- *Settings>> Save* – confirmation of changes to exported catalog; must be confirmed by clicking "*Save Settings"*
- *Help>> First Steps* – Start the programs help function
- *Help>> About* – Display window of information about the program

2. ***Tools from a given category*** – displays all tools and their parameters from a selected category, including: identifier, name, diameter, length, part number, type, and catalog number. The parameters of the selected tools can be seen in the "*Tool Details*" window.
3. ***Tool Picture*** – displays the tool's picture, or any file specified by the user. Each tool can have its own picture.
4. ***Technical Drawing*** – a field displaying a constant picture with the tool's parameters. Drawings are associated to a category of a given tool.
5. ***Other Parameters*** – a field that allows entering additional parameters other than those displayed in the "*Tool Details*" window. Available parameters are *materials, name of cut, type of cut,*. The *Comments* field is a text field and can store up to 1000 characters of arbitrary text.
6. ***Action Icons***:

- ***Add New*** – allows adding of a new tool to the category which the user is currently using
- ***Modify*** – allows modifying of the parameters of the current tool
- ***Save Changes*** – this becomes available after using the *Add New* or *Modify* buttons. This saves the changes to the tool database.
- ***To Toolbox*** – Button is active when the possibility to add a tool to the program toolbox is available. The toolbox allows the exporting of tools to the NX program.

7. ***Current Tool Details*** – this field displays all parameters of a tool of a given group. Certain fields were labeled by color based on the importance of the field.
8. ***Toolbox*** – saves parameters of tools that need to be exported. Tools in the toolbox cannot be repeated – the program will not allow adding of duplicate tools. The toolbox can store an arbitrary number of tools for any category.
9. ***Action Icons of the Toolbox***:
   Remove – removes a specific tool from the toolbox
   Export>>NX – exports all tools from the toolbox into a database file, which is read by the Unnigraphics system. The file is stored in a folder of the program settings *(Settings>>Export Catalogs).*

Frequently Asked Questions about "Technolog":

- ***How do I add a new tool?***

In order to add a new too, first select the category to which the tool belongs. The category selection is located on the side menu entitled "*Tool Categories*". Then click on the "*Add New*" icon, fill in the relevant parameters, and click "Save Changes"..

- ***How do I change tool parameters?***

Tool parameters can be modified by selecting the desired tool and clicking on the "*Modify*" icon.

- ***How do I change the image of a tool?***

After cliking on the" "*Modify*" icon, a new window will open. Click on the "Add" icon and follow the directions that appear on the screen.

The most important functionality of the "Technolog" application is the export of parameters to the NX system, as well as the import of parameters from an specified XML file.

To export information to the NX system, it is necessary to add it to the toolbox using the "*To Toolbox*" icon and then confirming the export using "*Export>>NX"* option. The tools and parameters will be exported to a database location specified by the user.

In order to import tools and parameters click on "*File>> Import from XML*". Next, select the XML file containing the tools and parameters that are to be imported into the system. Only one tool can be imported from one XML file. The integration of a tool from a file to the program is done via parameters that contain specific data. The list of parameters is located in the *importXML* column of the *names* database table.

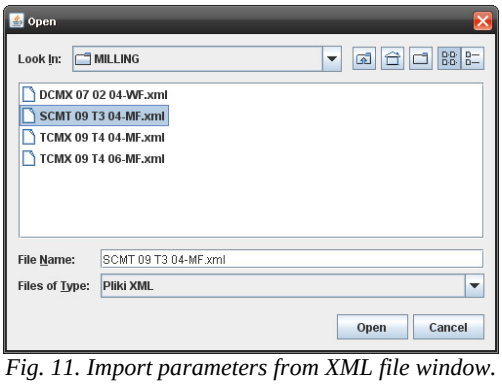


Fig. 11. Import parameters from XML file window.

Conclusion

The proposed application fully supports all requirements, such as storing, managing, and intgrating parameters with the technological tool database of the NX system.

The developed application delivers functionality that is lacking in the CAD/CAM system. It also presents the possibility of expansion of technology and programming tools. The application solves the problems of information exchange and information retrieval caused by repeated installations of the system. Satisfying the gathered requirements, the application provides functionalities such as importing, management, and exporting of parameters and tools of the NX system. Information among users and organizations is much more accessible.

The "Technolog" application has undergone tests to ensure its proper functionality. The program has been tested by 10 users of the NX system. The results were most satisfying. The results of the test show that the program has good usability and that all available functions are easily accessible. The learning curve is small. All

operations are performed rather intuitively, and the layout of windows increases readability and makes switching between functions easy.

There were no errors found while searching the technological database of the application. The tools were quickly displayed along with their parameters. Sorting was also working correctly and allowed for quickly locating desired tools. The export of tools also ran without errors.

The database is an innovative element to the CAD/CAM systems. It is very accessible across the industry.

Streszczenie

W pracy pokazano sposób tworzenia aplikacji dla systemów CAD/CAM. W języku Java napisano program pełniący funkcję gospodarki narzędziowej dla systemu NX. Omówiono podstawowe zagadnienia dotyczące technologicznych baz danych oraz dokonano klasyfikacji narzędzi skrawających: frezów, wierteł.

Słowa kluczowe: Technologiczne bazy danych, aplikacja, systemy CAD/CAM

Literatura

1. Barteczko K., Java wykłady i ćwiczenia, Mikom, 2000.
2. Parsian M., JDBC Metadata, MySQL, and Oracle Recipes. A Problem-Solution Approach, Apress, 2006.
3. Cichosz P. Narzędzia skrawające, WNT, Warszawa 2006
4. Eclipse - an open development platform, http://www.eclipse.org/ , 05.12.2008