Journal of Machine Engineering, Vol. 8, No. 4, 2008

CAD, technological equipments,
knowledge base, case based reasoning,
neural networks

Grigori NEKRASSOV[1]
Leonid PORTJANSKI[1]

KNOWLEDGE REPRESENTATION AND THEIR APPLYING TO FIXTURE DESIGN

The aim is to develop an integrated design space for CAD of technological equipments. The model of the design environment is represented as a two-level hierarchical decision-making system. The proposed environment provides support for creating and manipulating 3D models of technological equipment and their components, calculating functional and structural properties and the evaluation criteria, and for determining the rules to control the design process for optimizing the portfolio of interdependent projects of technological equipments. As examples the design of workholder is considered. The computer aided workholder design system is created on Case Based Reasoning, in which the attributes of the workpiece and structure of workholder as case index code are designed for the retrieve of the similar cases, and the structure and hierarchical relation of case library are stored. The structure of case storage (database and knowledge base) for general workholders selected as an object for mining and sampling of the developed software are worked out. It is offered to apply a method of neural networks to realization procedure of forecasting of decisions. The considered algorithms of knowledge extraction are realized by development of invariant nucleus design space.

1. INTRODUCTION

There are two ways of products' engineering:

- to realize engineering only using CAD system resources. Such of designing is implemented for original designs, required much time and great qualification of a designer;
- to realize engineering using the resources of preliminarily created environment of engineering for this CAD system. In this case the high qualification of a designer is required at the stage of designing the object-oriented environment, when is defined structure of a product, its geometrical characteristics and assembling rules. It takes less time and energy to execute the designing of specific parts of this group. It may be used methodology when creating environment of designing.

The Design Space (DS) model is based on the use of decomposition and categorization methods which represent for the given product family breakdown structure in terms of product functions (abstraction hierarchy) and relations between product components.

The DS model consists of:

- A feature-based representation (database - DB) of components of work-holders (WH)

[1] Tallinn University of Technology, Department of Machinery, Ehitajate tee 5, 19086 Tallinn, Estonia,
Grigori.Nekrassov@ttu.ee, Leonid.Portjanski@ttu.ee

(primitive and complex) which have its own syntax of description and semantic explanations and are described by geometric, dimensional, material etc. properties;

- Indexing method for retrieving from DB of WH's and their components, to stream line the matching process between new design task and existing solutions on the DB;
- A similarity metric to measure the similarity between a new desired work-holder or components and the existing in DB representation of previous designs and used compo nents.

The case base reasoning approach (CBR-tool) for a solution of problems is applied [1]. A CBR system solves new problem by adapting solution that were used to solve previous similar problems.

The DB of the DS holds for the given product family a number of cases each of which represents a problem description together with its corresponding solution. Once a new problem arises, a possible solution to it is obtained by retrieving similar cases from the DB according to the specification of the new design task and studying and if necessary modifying recorded solutions. Every time that a new design is solved a new experience could be retained and made available for future use. The process of retaining is controlled by the supervisor of the system. A very important feature of CBR is its coupling to learning. It denotes the use of a new computer-learning paradigm in design process. To divide the responsibilities and manage the CBR process for a design environment, the two level decomposition of DS was proposed in [2] and is used to develop a DS. The tasks of DB management and activities of learning from previous design are tasks of an upper or supervisory level.

The design task of a work-holder starts with a design problem description and includes the specification of the main features of new design problem. In the first step the previous design solutions similar to the new problem specification are searched from DB. If in DB exist no suitable WH model, the problem description is decomposed according to the abstraction hierarchies and the models of suitable components are retrieved. In praxis for most cases of WH design there are always some models suitable for use. In some cases new model must be designed using the capabilities of basic CAD system.

To show how the proposed technique might be useful, we describe a specific instantiation of this approach, in a prototype system for design of technological equipment (work-holders). The problems of work-holder design are a complex and highly experience-dependent tasks. A design environment for CAD of work-holders is characterized with the following features:

- A large solution space.
- A variety of input data for many knowledge sources.
- A multiple of company-specific standards, classification schemes etc exist.

2. MAIN PRINCIPLES OF MODELING OF AN INTEGRATED DESIGN SPACE FOR CAD

CAD process can be viewed as automated search in DB for those solutions that satisfy the requirements and are in some sense best among feasible alternatives. The ability to efficiently save, index, and retrieve alternative models has become critical in a wide range of applications, including CAD systems, indexing schemes for large component inventories, access methods for “smart catalogues,” and for performing searches through databases and on the Internet.

In solving the referred tasks we are interested in solving the following related problems:

- Given is a collection of models of WH and their components, it is necessary to estimate which ones are similar to each other and how similar are they?
- Given is a set of characteristics of new design task, what kinds of WH and/or its compo nents we are interested in, how to retrieve suitable models from a system DB?

During last years, a number of efforts have been made to develop algorithms to examine CAD designs and extract features that correspond to functionalities of designed product. Many classification schemes have been developed based on the idea, to capture critical features of a product in an alphanumeric string, for example the GT codes [5]. GT coding was intended to be human interpretable and has caused some difficulties in use these codes in CAD.

There are two main difficulties to solve the proposed task: dealing with the combinatorial explosion of the design space and handling large amount of domain dependent knowledge. To solve the proposed task a hierarchic classification schema of features is proposed.

The top node in the hierarchy is the description of characteristics of a product family. The bottom level entities are the multitude of models of components that could be used to construct the WH.

The components of a WH can be primitive (non-decomposable) or they may complex, consisting different subcomponents and corresponding structure.

The total classification tree for a product family could be represented by composing the general classification scheme for main functionalities of a product family and hierarchic classification schemas (abstraction hierarchies) of components. The structure of the functional hierarchy for WH consists [6]: clamping components, supporting components, fastening components, guiding elements, accessories etc. According to this approach a two levels of abstraction is used: the functional hierarchy and for each functional group of components the hierarchy of geometrical and dimensional information (abstraction hierarchy).

Abstraction hierarchies contain two types of relationships:

The “***type-of”*** (is-a) relationships that specify whether a call is a special case of another class. Abstraction hierarchy is an important issue of modeling the CAD domain data and knowledge. The main purpose of use the abstraction hierarchies, is to generate small groups of components that demonstrate similar characteristics. The type-of relationships help to decompose the DS into the subtasks and to estimate the similarity of components, to determine what data and knowledge should be applied.

Decomposition hierarchies described by the “***part-of”*** (consist) a relationship that specifies what components or parts are assembled into an assembly (component of higher level). Fig. 1. shows example of the substructure of work-holder domain with *Abstraction and Decomposition hierarchies.*

An important difference between these two hierarchies is that abstraction hierarchies do not change if the product family and class of work-holders (manufacturing operation) is not changed, whereas decomposition hierarchies vary with the changes in the situation of in the purchase of the components from different suppliers (related to the management of inventories).

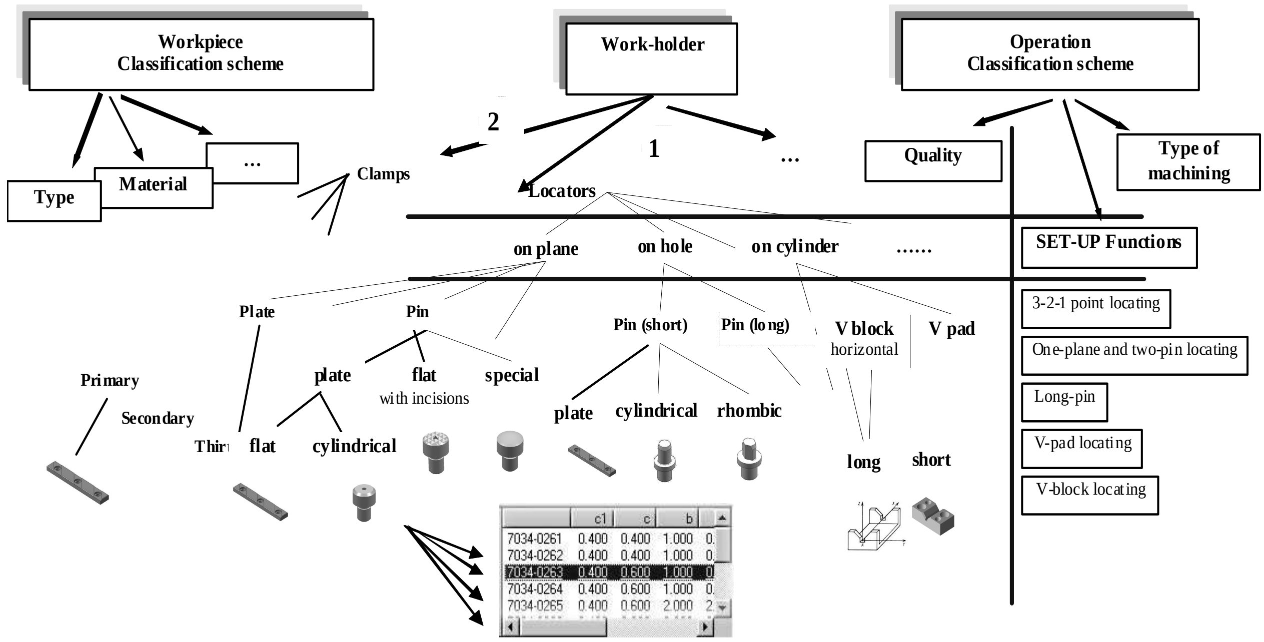


Fig. 1. The generic structure of the WH's components and their functional groups

To represent the problem domain we try to avoid the use of multiple inheritances to eliminate the problems of ambiguity related to the meaning of the attribute in a specialized class since it inherits the same attribute from multiple parents.

The most critical factor to measure the similarity in the work-holder design is the locating method. If the locating methods of two work-holder designs are the same, there is a basis for comparing their similarity.

Supporting access to the similar components through different classification schemas raises a number of issues. The classification schemas are specialized, and therefore a new user may not be familiar with the schema or terms that are employed. To solve this problem the corresponding computer support is needed.

The main guidelines for representing the information content of the DB are the following:

- Small number of product families is recommended to use, in order to generate DS model with consistent number of components.
- Assignment to each class of components a few significant features and attributes.
- Adding capability to visualize the component 3D models, to have an immediate idea of its usefulness.
- The of flexible classification structures, to adapt to continuous changes of the situation in a company.

3. CBR IN CAD

The cases are usually stored in databases in free, unstructured format, and they can be even distributed among many databases. The structure of representation can be dynamically changed in CBR process and may involve different data format for representation of the value of the features. Use of the proposed general similarity concept creates the possibility to compare cases that have different structure representation in the case base. Moreover this representation can be even built dynamically before retrieval.

CBR system is dynamic in the sense that, cases representing new problem together with their solutions are added to the case base, redundant cases are eliminated and others are modified by combining existing cases. CBR is an incremental learning approach because every time that a problem is solved a new experience can be retained and made immediately available for future retrievals. Design knowledge accumulation and storage in case base of the system is critical in design activities.

Much time and effort is spent on applying the methods of information systems technology and knowledge engineering effectively in the projects of implementing CAD systems.

To enhance the effectiveness of methods for each of real CAD project we need to adapt knowledge representation and extraction methods or construct the new ones so that they can fit to the real product development environment in enterprises.

In the design space development the problem arises how to decompose the design process so that individual design tasks are communicating only over the supervisory subsystem [2].

Wizard is the main element of environment, supporting to design objects of this group.

This report shows us the possibility of creating the wizard master, which is invariant to designing objects. The scheme of wizard master's work is corresponded with CBR methodology - description of a problem, 4-RE cycle performance (retrieve, reuse, revise, retain) [1], with the need of the problem's decomposition with its specification performance and etc. Creating environment of designing with using wizard master is implemented on supervisor- level.

Tooling wizard master to the given group of tools is made with the help of worked out language of high level as well as CAD system means. In the latter case a designer puts together goals from parts, pointing out conditions of impurities or a dialogue's call for choosing the properly element of a design. In that way the script of a designing process is created. This script is and/or a graph. There are procedures at the top of a graph which is necessary for designing of this type of parts; there are terms of procedures' choosing on arches. It is determined, that the least developed is the problem of construction of procedure of forecasting of decisions invariant to objects of designing. It is offered to apply a method of neural networks to its realization.

The "Design Reuse" aims to maximize the use of successful past designs in part design and in whole for new design.

CAD process can be viewed as automated search in design space (DS) for those solutions that satisfy the requirements and are in some sense best among feasible alternatives. The model of design space is rather complex, it reflects the complicated inner links in domain knowledge.

The ability to efficiently save, index, and retrieve models of typical solutions, components etc in DS has become critical in a wide range of CAD systems applications, including indexing schemes for large component inventories, access methods for “smart catalogues,” and for performing searches through databases and on the Internet.

In solving the referred tasks we are interested in solving the following related problems:

- given a set of characteristics for what kinds of products and components we are inter ested in, how to retrieve suitable models from a system database;
- given a collection of models of products and components, it is necessary to estimate which ones are similar to each other and how similar are they;
- management of the content of the case-base to restrict the uncontrollable increase of the case base.

To show how the proposed technique might be useful, we describe a specific instantiation of proposed approach in a prototype system for design of technological equipment. Technological equipment design is a complex and highly experience-dependent task.

Nowadays, it is necessary to integrate different software tools that are used during the design process in order to efficiently support design teams.

CAD systems require in many cases dynamic solutions and learning and adaptation mechanisms. A CBR system can be used by itself or as embedded part of a CAD system. CBR systems are especially appropriate when the rules that define a design knowledge domain are difficult to obtain or the number and the complexity of the rules affecting the design problem solution are too large.

Computer aided design of technological equipment is a promising application field for case-based reasoning. The fact is that much of equipment (fixture, die, etc) designs are not very different from what has been done in the past.

CBR does not require the obvious model of a data domain. The given model is shaped in process of accumulation of knowledge of the given data domain, that largely easies process of eliciting of knowledge being extreme in implementation of model and requiring of high proficiency of the implementations and occupying many years. The process of definition of a similar decision is reduced to identifying present arguments.

But for a successful operation first of all it is necessary to determine:

- an indispensable and sufficient list of the attributes describing the given process which is capable to determine from a present knowledge base of decisions - a similar decision.
- to elaborate dynamically modified classifiers of data of arguments,
- to use methods of the database for a storage of information,
- to foresee a process of self training with acquisition of new knowledge.

One of the major stages of applying CBR is development the architecture of case library. The difference of the data base of the previous solutions from case library is indexation of these solutions.

A case library can be organized in various forms such as a linear list of cases, a hierarchy of classes of cases and other forms. In the design of a case library, three important issues must be addressed before any implementation is attempted; they are case index, case representation, and retrieval strategy.

The case library consists of a number of predefined cases. Case representation is one of the most important issues in case-based reasoning. Any case in the case library is composed of design feature description and the design space model. The building a qualitative DS model is very important.

The DS model is based on the product decomposition which represents the product family breakdown structure in terms of product function and entities (product components). Such a model must provide a representation of the products in a variety of forms to satisfy the needs of different users.

The next step is to create the 3D parametric models of products and their standardization components using instruments available (the kernel system – SolidEdge and MechSoft). The example of description of variables is shown in Fig. 4. All the external variables are represented and stored in Excel tables. Different types of 3D models of parts are realized using the abilities of SolidEdge to suppress features and to store new types of models in the library.

When 3D models products are created and included in the database, it is necessary to use "Group manager" of MechSoft system to join these components in the functional groups. It is necessary to make a link between the dimension difference modules (parts) as shown in Fig. 2, in which we can also see the tree of a *G-clamp* assembling and decomposition of this clamp unit.

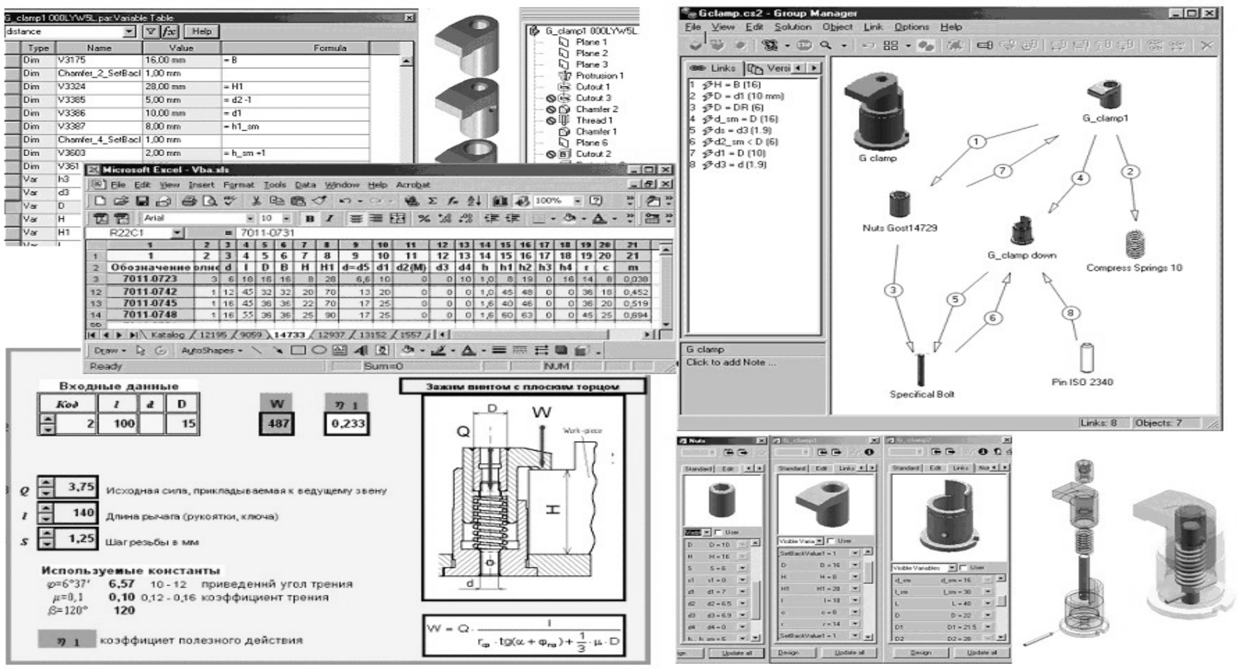


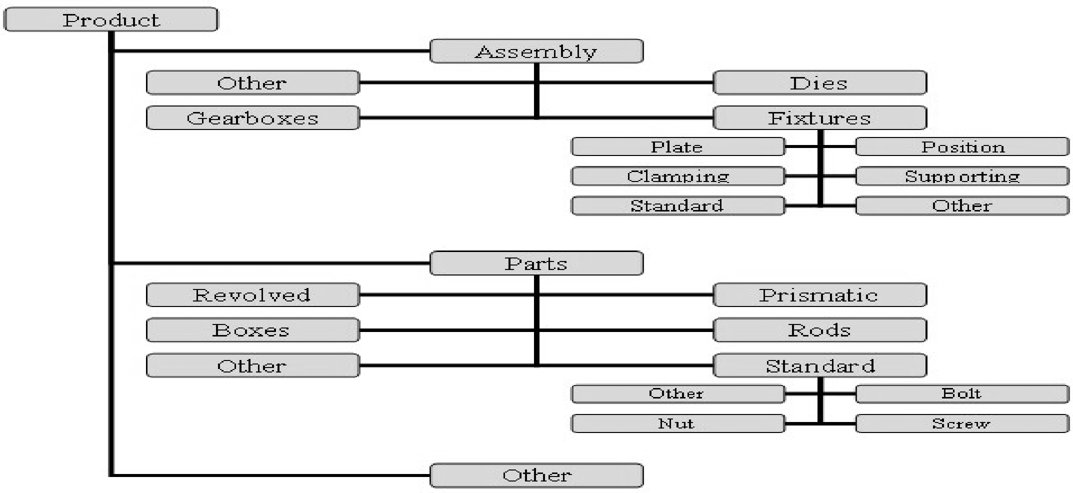
Fig. 2.3D parametric module (G-clamp) generalized for group of similar components and Force calculation

The knowledge base contains different information about products (such as for fixture - basic locating methods (datum), clamping methods end et), programs of engineering and economic calculations, all production data components and units of work-holder, such as product cost, quantity, unit, supplier, availability, frequency, and etc.

To solve the task of description a DS, a *hierarchic decision tree (schema) of feature-based component representation* is proposed. The root node represents the initial problem definition, a general description of a product family. A child node can be obtained from its parent node (functional or geometric feature) through a specifying the additional properties or functions. The bottom level entities are the multitude of components that could be used to construct the product.

The components of a product can be primitive (non-decomposable) or they may consist different subcomponents and corresponding structure.

To avoid the explosion of the number of decision nodes (states) occurring when we specify the DS of a complicated CAD system, two types of structuring techniques for are adopted: one is called ***AND decomposition*** for concurrency, and other one is ***OR decomposition*** for clustering of alternative solutions (see Fig. 3).



Assembly	OR	Fixtures	Geaboxes	Dies	Other			
Fixtures	AND	Plate	Position	Clamping	Supporting	Standard	Other	
Parts	OR	Revolved	Prismatic	Boxes	Rods	Standard	Other	
Standard	OR	Bolt	Nut	Screw	Other			
Plate	OR	T_slotted	Threaded	Other				
Position	OR	....	......	....	Knife edge	Other		
Clamping	OR	....	......	....				
Knife edge	Part	**Name**	**d part**	**H**	**L**	**B**	**B1**	**d**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Name	**d part**	**H**	**L**	**B**	**B1**	**d**	**d1**	**d2**	**A**	**A1**	**A2**	**l**	**h**	**h1**	**b**	**r**	**c**	**D**	**L1**
7033-0033	>15 and <20	25	16	48	18	6,5	11	5	32	9	12	8	6,5	9	6	1	0,6	18	28,73
7033-0031	>5 and <10	16	10	32	8	4,5	7,5	4	20	6	7,5	5	4,5	5	2	0,6	0,6	8	15,66
7033-0032	>10 and <15	20	12	38	14	5,5	9,5	4	26	8	10	6	5,5	7	4	1	0,6	12	19,49
7033-0033	>15 and <20	25	16	48	18	6,5	11	5	32	9	12	8	6,5	9	6	1	0,6	18	28,73
7033-0034	>20 and <25	25	20	55	24	6,5	11	5	40	9	12	8	6,5	11	8	1	0,6	22	34,56
7033-0035	>25 and <35	32	25	70	32	9	14	6	50	12	16	10	8,5	14	12	1,6	1	30	45,21
7033-0036	>35 and <45	40	32	85	42	11	17	8	63	16	20	12	11	18	16	1,6	1	40	59,28
7033-0037	>45 and <60	40	38	100	55	11	17	8	76	16	20	12	11	22	20	1,6	1	50	70,86
7033-0038	>60 and <80	50	45	120	70	13	19	10	95	22	26	14	13	28	25	1,6	1,6	70	94,5
7033-0039	>80 and <100	55	50	140	85	13	19	10	112	27	30	14	13	32	32	1,6	1,6	90	116,14

Fig. 3. Fragment of knowledge representation

4. CASE INDEXING

The choice of attributes for feature description is one the most critical phases. The representation of design cases requires various models because design content involves topological, geometric, and physical properties and relations between them. The eliciting of an indispensable and sufficient list of the attributes and also their classification is the most laborious problem which required high proficiency of the designer in the given data domain. With problems, bound with designing part and their manufacturing by most relevant on our view are: classification of parts and assemblies, a material of parts, fidelity of fabrication, critical dimensions, the cost, the supplier etc.

Many classification schemes have been developed based on the idea, to capture critical features of a product in an alphanumeric string, GT code. GT coding was intended to be human interpretable and has caused some difficulties in use these codes in CAD.

Some systems [3] used from 6 up to 16 digits to compose the index code that included workpiece shape, machine portion, bushing, locating device, clamping device, material and etc. But this system cannot be used for other types of part. The total classification tree for a family of products could be represented in step-by step manner by composing the general classification scheme for main functionalities of a product family and trees of components.

Two levels of abstraction are proposed: the *functional hierarchy* and for each functional group of components the *constructional* hierarchy of geometrical information.

The case index should be composed of all features of the parts from different domain area and dynamically changed. Therefore, we proposed following structure of index.

We propose to use index separately for each attribute according to its structure. 1/3/1/…/2

Where: / is a separator of levels of structure, and digit - number of a set in the given level. These index are connected to file of part and at once are filter by operation CBR for selection of similar case. For the assemble user can see list of all parts.

Besides, an administrator of a system has capability to change structure of attribute (add new alternatives of a set). On the bottom level the list of specification statement parametric data of a part is introduced, which one is connected to the data table for constructing family part in CAD.

5. NEURAL NETWORKS

Neural networks is the group of analytical methods, based on (hypothetical) principles of teaching intellectual beings and brain functioning and premising to forecast values of some variables in new observations according to other data of observations after passing the teaching stage according to existing data [8].

Neural networks are one of methods of so-called extraction of the data.

Last years huge databases in which the information of scientific, economic, business and political character is stored are created. For job with similar bases the computer technologies are developed, allowing keeping, sort and visualizing the data, to provide fast access to them, to carry out their statistical processing. Considerably achievements in development of methods and the programs, capable to find out in the data important, but the latent information is smaller, however. Similar *not trivial* extraction *implicit, before the unknown person and potentially the helpful information* from the big databases also are called as Data Mining or Knowledge Discovery. We shall use further for the description of this area of computer science term - knowledge extraction.

Knowledge extraction uses the concepts developed in such areas as Machine Learning, Database Technology, statistics and others. Job with very big databases demands efficiency of algorithms, and discrepancy and, frequently, incompleteness of the data generate additional problems for extraction of knowledge. Filling of misses in databases is one of the problems solved by neural networks. However, the main claim to neural networks always was absence of an explanation. Extraction of rules from neural networks means their preliminary training. This procedure demands a lot of time for the big databases.

Let's consider one of methods of extraction of rules from the neural networks, trained to the decision of a problem of classification [4]. This method refers to **NeuroRule.**

The problem will consist in classification of some data set with the multilayered perseptron and the subsequent analysis of the received network with the purpose of a finding
of the classifying rules describing each class.

6. ALGORITHM OF KNOWLEDGE EXTRACTION

Let A designates a set from N properties $A_1,A_2,...,A_n$, and {a_i}- set of possible values which can accept property A_1.We shall designate through C set of classes $c_1,c_2,...,c_m$.

For training sample the associated pairs vectors of entrance and target values $(a_1,...,a_m,ci)$, where $c_i \in C$.

The algorithm of extraction of classifying rules includes three stages:

- **Training of a neural network.**

On this first step two-layer perseptron trains on a training set down to reception of sufficient accuracy of classification. We shall assume that the training data set is necessary for classifying on two classes A and B. In this case the network should contain N the entrance and 2 days off neurons. To each of classes there will correspond the following of activity of the days off neurons (1,0) and (0,1). The suitable quantity neurons in an intermediate layer, generally speaking, cannot be defined beforehand - their too big number conducts to conversion training while small does not provide sufficient accuracy of training. All methods of adaptive search of number neurons in an intermediate layer share on two classes, according to that, from small or big number intermediate neurons will start algorithm. In the first case, in process of training in a network are added additional neurons, in opposite - after training there is a destruction excessive neurons and connections. NeuroRule uses last approach so the number intermediate neurons get out big enough. We shall notice that NeuroRule destroys as well superfluous entrance neurons which influence on classification is not enough. As function of activation intermediate neurons the hyperbolic tangent so their conditions change in an interval [-1, 1] is used. At the same time, function of activation of the days off neurons is Fermi's function (conditions in an interval [0, 1]).

- **Filtration of a neural network.**

The trained neural network contains all possible connections between entrance neurons and neurons on the latent layer, and also between last and target neurons. The full number of these connections is usually so great, that from the analysis of their values it is impossible to take foreseeable for the user classifying rules. Filtration consists at a distance excessive connections and neurons, not resulting to increase in a mistake of classification by a network. The resulting network usually contains a little neurons and connections between them and its functioning gives in to research.

- **Extraction of rules.**

Even if the parameters describing attributes of classified objects represent continuous sizes, for their representation it is possible to use binary neurons and a principle of coding such as the thermometer. At such way of coding the area of change of parameter shares on final number of M of intervals and for representation of all values lying in *m* interval the following condition of M binary neurons is used

At this stage from a neural network the rules are taken having the form

$$if\ \{a_1 \Theta q_1\}\ and\ u\ (\ a_2 \Theta q_2)\ and\ ...\ and\ (\ a_n\ \Theta\ q_n)\ then\ c_j.,$$

where $q_1,...,q_n$ constants,

Θ - the operator of the relation (=> <,> <)

It is supposed, that these rules are obvious enough at check and are easily applied to the big databases.

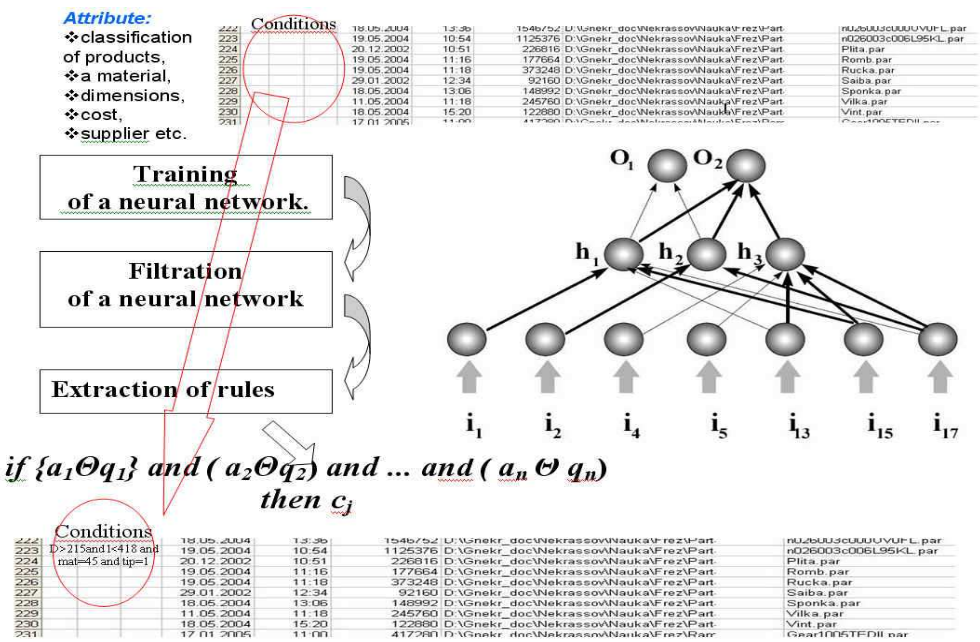


Fig. 4. Extraction of rules

Let's mention even one procedure which is carried out at extraction of knowledge from neural networks - correction. Similar operation has been offered Weigend and colleagues and as a matter of fact is used in parallel with training [9]. Hybrid use of training and correction of the data carries name **CLEANING** (CLEARING+LEARNING). The given procedure includes ascending process of training at which the data change connections in a neural network and descending process in which the neural network changes the data on which training is made. The offered technique allows to form „if then" rules dynamically (Fig. 4).

7. CONCLUSION

The proposed approach is intended to be general in the sense that the same basic ideas and could be used in different engineering design application domains. The purpose of this approach is to construct a reasoning model, which is applied to the evaluation of mechanical designs. General mechanical design systems always adopt a design-evaluation-redesign procedure. Analysis and evaluation are required to conclude whether a design concept is satisfying or not, so evaluation is an indispensable part in intelligent CAD systems.

However, the information represented for design, as well as the criteria for judging the similarity of design is heavily domain-specific. As an example of a particular application domain, we have focused on fixture design.

In the development of design space for fixture design the greatest difficulties arise at a formulation of conditions of a choice of decisions. The information received from qualified experts, should be written down directly in programs. In this case it is static and does not take into account accepted decisions at operation design space for fixture design. Application a neural network allows not only to predict the accepted decision, but also to take knowledge from databases.

The offered technique allows to form „if then" rules dynamically. The considered algorithms of knowledge extraction are realized by development of invariant nucleus design space for fixture design. The neural network is a black box to which can ask questions and to receive from it answers and explanations.

REFERENCES

[1] AAMODT A., PLAZA E., *Case-Based Reasoning: Foundational Issues, Methodological Variations, and system Approaches*. Artificial Intelligence Communications Vol. 7 Nr. 1, March 1994, 39-59.

[2] KYTTNER R., NEKRASSOV G., SUTT A., *Design space for collaborative concurrent CAD of technological equipment*, DAAAM International Scientific Book ISBN 3-901509-36-4, Vienna, Austria, 2003, 347-358.

[3] LEE K. S., LUO C., *Application of Case-Based Reasoning in Die-Casting Die Design Adv*. Manuf. Technol.V 20, 2002, 284–295.

[4] LU HONGJUN., SETIONO R., LIU HUAN., *NeuroRule: A connectionist approach to Data Mining*. Proc.ofthe 21st VLDB Conference, Zurich, Swizerland, 1995.

[5] MITROFANOV S. P., *The Scientific Principles of Group Technology*. National Lending Library, Translation 1966.

[6] RONG Y., ZHU Y., *Computer-Aided Fixture Design*, Marcel Dekker, Inc. ISBN: 0-8247-9961-5, USA,1999.

[7] TIWARI M.K., RAMA KOTAIAH K., BHATNAGAR S., *A Case-Based Computer-Aided Process-Planning Sys tem for Machining Prismatic Components Adv*. Manuf. Technol. V. 17, 2001, 400–411.

[8] WASSERMAN P.D., *Neural Computing: Theory & Practice*. Van Nostrand Reinhold: New York. ISBN 0-442-20743-3, 1989.

[9] WEIGEND A.S., ZIMMERMANN H.G., NEUNEIER R., *Clearing. In Neural Networks in Financial Engineering*, World Scientific, Singapore, 1996.