ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992
Seria: MECHANIKA z. 108 Nr kol. 1161

International Conference on
COMPUTER INTEGRATED MANUFACTURING
Internationale Konferenz über
RECHNERINTEGRIERTE FERTIGUNGSSYSTEME
Zakopane, March 24-27 1992

Ryszard WYCZÓŁKOWSKI
Jan KAŹMIERCZAK

Chair of Fundamentals of Machine Design
Silesian Technical University, Gliwice, Poland

AN APPLICATION OF AN EXPERT SYSTEM FOR THE NEEDS OF AIDING THE CHOICE OF ALGORITHMS IN ENGINEERING CALCULATIONS

Summary. The paper introduces the shell expert system VV_SHELL and shows a possibility to use this system as the tool of aiding engineering calculations in the process of machine designing. The calculations of basic parameters of gear train are used in the paper as the particular example which illustrates the proposed solution.

1. Introduction

The engineering calculations are the one of these elements in the design process which can be effectively supported by means of computer technique. In majority of cases the computer programs, written for such a purpose and covering a particular part of the calculations (i.e. using a fixed algorithm) have strictly defined forms of input and output data. On the other hand the needs of users may be various. In particular, in some cases the same values can form the known subset of data but in other cases the same data have to be found in result of calculations.

Certainly it is possible to vary the input set of data for each program by means of adding new subroutines to it. In such a case the author of the program fixes 'stiffly' the algorithm of computations which are carried on when the program runs. Such a fixed algorithm is 'stiffly' written into the structure of the program.

Anyway, if an author of the computer program (as well as every people who makes calculations using a pencil and a piece of paper) is able to choose for each particular task the routine which gives an acceptable solution of the problem, thus it is probably possible to generalize the method of such a choice. If so, we would be able to use such a generalized approach for the cases when for instance the big number of input variables is used and a lot of possible paths of solving the problems has to be considered. Following this mind we can formulate the problem focusing on development of such a computer program, which would be able itself to find an optimal algorithm of computations on the base of fixed input data and an expected form of the output.

For the needs of such a solution the knowledge about contradistinctions between the features of the subject of calculations should be properly described and the manner of moving on the net of these contradistinctions should be identified, too. Such a task can be put, among others, by use of programs from the family of 'expert systems'. In this paper one of such systems is introduced. The shell expert system called 'VV_SHELL' has been developed in the Chair of Fundamentals of Machine Design by Dr Wojciech Cholewa, Professor of the Silesian Technical University in Gliwice. Prof.Cholewa has also formulated the ideas of utilizing the VV_SHELL system in some particular areas (among others in the field of CAD) and created algorithms which may enable the system to be used effectively in many cases.

2. Structure of the VV_SHELL system

There is a number of possible solutions of programs which belong to the family of the 'expert systems'. The most significant feature of all these programs is a manner of representing, storing and processing knowledge which is used in a particular system.

In the VV_SHELL the knowledge is stored in the form of **frames**. The frame describes a selected object or a group of objects and its structure contains a number of fields called **slots**. Every slot contains **facets** which are storing **descriptors** of the object/objects covered by the frame. The values of the features of the object or the connections between this object and other ones may be treated as the descriptors. The frames, slots and facets are identified by means of names so that in one frame two slots with the same name and in one slot two facets with the same name can not exist.

The VV_SHELL system has some particular features of the above mentioned elements of its structure. In every slot the facets of special destination may appear which are signed as *value*, *if_added*, *if_needed*, *if_removed*. Value of the facet *value* is understood as the value of the slot. If such a facet doesn't exist then the facet of the name *if_needed* will be looked for which should contain the description of the proper subroutine (the so-called **demon**). The value which is given back as the result of running this subroutine is treated as the value of the slot. The facets *if_added* and *if_removed* should contain the subroutines which are run for the purpose of adding something to the contents of the facet *value* or removing the contents of it, respectively.

For every frame the list of higher-order (superior) frames may be determined. The list of frames which are directly superior to the considered frame is contained in the slot with the name ***ako*** in this frame. The lack of such a slot means that for the considered frame no superior frames have been determined. The contents of the facets of the superior frames in the VV_SHELL system are inherited. If the sought facet doesn't exist in the slot of the given frame or if the demon contained in it can not be run then the seeking will be continued in the superior frame.

In order to use the VV_SHELL for a particular purpose, i.e. for carrying on calculations the only thing to do is to complement the knowledge base. The complementation of the knowledge base in the VV_SHELL system consists on recording the proper frames. The knowledge base in the system can be divided into two parts which concern two types of frames:

1) frames which contain information about the subject of activity (referring to decisive tables)
2) 'organizing' frames which utilize information from the first part for the needs of controlling the run of action (for instance - the run of calculation)

The organizing frames, except the controlling of the run of calculations, have to protect the system against a 'kink'. The mechanism of such a kink can be easy illustrated by use of the simple example of an equation with two unknown values x and y, for instance:

$$x * y = 12 \qquad (1)$$

From the equation (1) one can find values of x and y:

$$x = 12 / y \qquad y = 12 / x \qquad (2)$$

Let's try to find the value of x if the value of y is unknown in the following manner:

Step 1. It is checked if the value of x is known.

Step 2. Because the value of x is not known, the procedure of calculating this value is run.

Step 3. Because the value of x might by calculated only by use of the equation $x = 12 / y$ than it will be checked if the value of y is known.

Step 4. Because the value of y is not known, the procedure of calculating this value is run.

Step 5. Because for the needs of calculating the value of y it is necessary to know the value of x ($y = 12 / x$) the try to calculate of x will be undertaken again and thus the calculation would be kinked (coming back to the step 2).

The organizing frames in the VV_SHELL can protect the expert systems against such a kind of effects.

3. Description of the proposed procedure

After introducing the main features of the proposed tool we have to describe the procedure of working of the system in the intended kind of action. In particular we want to supply the calculation of some values (the values of constructional features) in the case when the algorithm of the calculation is not given 'a priori'. The proposed procedure bases on answering the question concerning the existence of a values of some sought quantity. The proposed algorithm might be described as follows:

Question: Do you know a value of **X**?

Activities undertaken:

Step 1. Check whether the value of **X** does exist.

Step 2. If it does then answer YES and terminate the action.

Step 3. Try to fix the value of **X** according to the procedure attributed to the **X**.

Step 4. Check once again whether the value of **X** does exist (in a case when you are not able to fix the value of **X** in the step 3), answer YES or NO (agreed with the ascertained state) and terminate the action.

It can be seen the initial assumption has been accepted in the introduced procedure that the existence of each value is always checked before executing any operation with this value. This assumption has to be taken into consideration when the computational program is realized.

In order to explain the introduced algorithm in detail the example is used in this paper of fixing the following basic parameters of gear train:

- pitch diameter of the small gear d_{p1}
- pitch diameter of the big gear d_{p2}
- number of teeth of the small gear z_1
- number of teeth of the big gear z_2
- frontal module pitch m_t
- transmission ratio i
- interaxial distance a_w

As the base of further operations the following dependencies have been utilized:

$$d_{p2} = i * d_{p1} \quad (3)$$

$$d_{p1} = z_1 * m_t \quad (4)$$

$$d_{p2} = z_2 * m_t \quad (5)$$

$$a_w = \frac{1}{2}(d_{p1} + d_{p2}) \quad (6)$$

The dependencies shown above have been transformed to get the formulas which tie every one of the parameters with the rest of them, i.e.

$$d_{p1} = \frac{d_{p2}}{i}; \quad d_{p1} = m_t * z_1; \quad d_{p1} = 2 * a_w - d_{p2}; \quad (7)$$

$$d_{p2} = d_{p1} * i; \quad d_{p2} = m_t * z_2; \quad d_{p2} = 2 * a_w - d_{p1}; \quad (8)$$

$$i = \frac{d_{p2}}{d_{p1}}; \quad i = \frac{z_2}{z_1}; \quad (9)$$

$$z_1 = \frac{z_2}{i}; \quad z_1 = \frac{d_{p1}}{m_t}; \quad (10)$$

$$z_2 = z_1 * i; \quad z_2 = \frac{d_{p2}}{m_t}; \quad (11)$$

$$m_t = \frac{d_{p1}}{z_1}; \quad m_t = \frac{d_{p2}}{z_2}; \quad (12)$$

$$a_w = \frac{d_{p1} + d_{p2}}{2};\qquad (13)$$

These dependencies can be written into the following table:

Tab.I.

	Table Td1	R1	R2	R3
C_d2	Is the value of d_{p2} known?	Y	-	Y
C_aw	Is the value of a_w known?	-	-	Y
C_z1	Is the value of z_1 known?	-	Y	-
C_i	Is the value of i known?	Y	-	-
C_mt	Is the value of m_t known?	-	Y	-
A1	$d_{p1} = d_{p2} / i$	*		
A2	$d_{p1} = z * m_t$		*	
A3	$d_{p1} = (2 * a_w) - d_{p2}$			*

The table introduces two categories of the frames:

- the frames containing information about the subject of calculations (which refers to the 'decisive' frames like Td1)
- the organizing frames which utilize the information from the frames of first category for controlling the run of calculations.

The frames used in the example are shown below. The name of every frame of the first category has the form C_xx and relates to the quantity which is described by it (for example: C_d_{p1} describes the pitch diameter of the small gear). These frames are giving back logic values:

- *YES* if the value of *xx* is known
- *NO* if the value of *xx* is unknown

The frames of the form **Rx** describe rules: If the condition contained in the slot *if* is fulfilled run action described in the slot *then*.

The frames of the form **Ax** describes the actions which are undertaken by the system.

In our example the following frames are utilized:

- **FRAME d1**
  SLOT ako
  value=(Obiekt)
  SLOT table value = (R1 R2 R3)
- **FRAME C_d1**
  SLOT ako value = (_ST)
  SLOT return if_needed = (VIEW_FIX (d1 _YES))
- **FRAME R1**
  SLOT ako value = (_RU)
  SLOT if value = (C_i C_d2)
  SLOT then value = (A2)
- **FRAME A1**
  SLOT ako value = (_AC)
  SLOT return if_needed = (SET (DIV (VIEW_FIX (d2 _WARTOSC))
  (VIEW_FIX (i _WARTOSC))) (d1 _WARTOSC))
- **FRAME R2**
  SLOT ako value = (_RU)
  SLOT if value = (C_mt C_z1)
  SLOT then value = (A1)
- **FRAME A2**
  SLOT ako value = (_AC)
  SLOT return if_needed = (SET (MULT (VIEW_FIX (mt _WARTOSC))
  (VIEW_FIX (z1 _WARTOSC))) (d1 _WARTOSC))
- **FRAME R3**
  SLOT ako value = (_RU)
  SLOT if value = (C_aw C_d2)
  SLOT then value = (A3)
- **FRAME A3**
  SLOT ako value = (_AC)
  SLOT return if_needed = (SET (SUB (DIV (VIEW_FIX (aw _WARTOSC)) 2)
  (VIEW_FIX (d2 _WARTOSC))) (d1 _WARTOSC))

The organizing frames have to state if the value of a given quantity is known (frames C_xx) and if no the proper action is run in order to fix this value. The controlling of this action consists in checking the rule conditions from the list contained in the slot *table* as long as a rule with the fulfilled condition is found and then the action is undertaken according to the contents of the slot *then* in this rule.

When we are seeking for instance the value of the pitch diameter d_{p1} by use of the proposed algorithm the following actions are carried out:

Step 1. **It is checked if the value of d_{p1} is known.**

Step 2. Because d_{p1} is not known the routine of seeking this value is run on the base of the list of rules which are contained in the slot *table* of the frame C_d1.

Step 3. The condition (first in the list) of the rule R1 is checked.

Step 4. Because the value of i is not known the try to fix this value is undertaken.

Step 5. The value of i is calculated by use of the dependence $i = z_2 / z_1$.

Step 6. The condition of the rule R1 is accepted (known d_{p2} and i) and thus the value of d_{p1} is found by use the dependence $d_{p1} = d_{p2} / i$.

The particular algorithm which brings to fixing the needed value is chosen by the system itself. The choice depends basically on input values. 'By the way' the program calculates other values (for instance the value of i in the example shown above) but only these ones which are necessary for the purpose of calculating the sought value.

4. Advantages and disadventages of the proposed solution

The simplicity of building the calculating program as well as facility of modifying and extending the program are undoubtly important advantages of the tool introduced in this paper.

If the organizing frames still exist the author of the particular program which bases on the VV_SHELL system will have only to fulfil the base of knowledge about the subject of calculations. Because the form of the 'informative' frames is close to the 'traditional' form of decribing rules (*if* <*A*> *then execute* <*B*>) and for the purpose to describe a quantity the only thing to do is to record the list of rules which enables one to fix this quantity so that the creation of programs by use of the VV_SHELL should not make troubles. The eventual extension of the computational abilities of the program is done by writing in additional rules (the organizing frames don't must be changed).

On the other hand the lack of the 'stiffly' fixed algorithm is a disadvantage of the introduced solution because it makes difficulties in the tracking of the run of calculations. By the way it makes problems with finding errors in the program because, even if we know in what frame the program actually 'stays', the particular way of getting this place can not be strictly determined. In the VV_SHELL a system of searching the execution of rules is built in which is added to the organizing frames.

5. Examples of the dialogue with a user

Example A. The input set of data contains the proper number of values which are needed for the purpose to fix the value of the searched quantity.

The tracking of the run of reasoning is switched on.

The searched quantity: pitch diameter $\mathbf{d_{p1}}$

Set of input data: $d_{p2} = 99$; $z_1 = 21$; $z_2 = 63$

** For the class _RU the DEBUG is switched on.
** DEBUG: checking the condition IF() in the rule R1.
** DEBUG: checking the condition IF() in the rule R10.
** DEBUG: the condition IF() in the rule R10 has been accepted.
** DEBUG: the action described by the rule R10 has been done.
** DEBUG: the condition IF() in the rule R1 has been accepted.
** DEBUG: the action described by the rule R1 has been done.

Result of calculations: $\mathbf{d_{p1} = 33}$

The frames which were utilized for the needs of fixing the value of d_{p1} are shown below. The contents of the frames Axx is omitted and only the action of these frames is marked in. The frames C_xx check the existence of the value 'xx' and run routines which fix this value in a case of lack of it.

- **FRAME d1**
  SLOT table value = (R1 R2 R3)
- **FRAME d2**
  SLOT table value = (R4 R5 R6)
- **FRAME mt**
  SLOT table value = (R10 R11)
- **FRAME z1**
  SLOT table value = (R12 R13)
- **FRAME z2**
  SLOT table value = (R14 R15)
- **FRAME R1**
  SLOT if value = (C_mt C_z1)
  SLOT then value = (A1)
- **FRAME A1**
  /* d1 = mt*z1 */
- **FRAME R10**
  SLOT if value = (C_d2 C_z2)
  SLOT then value = (A10)
- **FRAME A10**
  /* mt = d2/z2 */

Example B. The input set of data contains not enough values to calculate the searched value.

The tracking of the run of reasoning is switched off.

1991-09-23 11:52

The searched quantity: pitch diameter $\mathbf{d_{p1}}$

Set of input data: $d_{p2} = 99$; $z_2 = 63$

THERE IS NOT ENOUGH DATA TO FIX d_{p1}

The dialogue:
Czy znasz wartosc (do you know the value) -> d_{p1}
nie znam (I don't)
Czy znasz wartosc (do you know the value) -> z_1
Podano wartosc (the value was given) **21**

Result of calculations: $\mathbf{d_{p1} = 33}$

Example C. No input data.

The tracking of the run of reasoning is switched off.

1991-09-23 12:00

The searched quantity: pitch diameter $\mathbf{d_{p2}}$

Set of input data: empty

THERE IS NOT ENOUGH DATA TO FIX d_{p2}

The dialogue:
Czy znasz wartosc (do you know the value) -> d_{p2}
nie znam (I don't)
Czy znasz wartosc (do you know the value) -> m_t
Podano wartosc (the value was given) **3**
Czy znasz wartosc (do you know the value) -> z_1
Podano wartosc (the value was given) **21**
Czy znasz wartosc (do you know the value) -> z_2
Podano wartosc (the value was given) **63**

Result of calculations: d_{p2} = 189

The research work which is introduced in this paper has been supported partially by means of the research project PB-132/9/91/KBN Poland.

REFERENCES

1. Cholewa W., Pedrycz W.: "Systemy Doradcze", Skrypt Politechniki Śląskiej nr 1447. Gliwice 1987.
2. "Szkieletowy System Doradczy W_SHELL - Instrukcja Użytkownika." red.: W. Cholewa. Raport nr G-547/RMT-4/91. Instytut Mechaniki i Podstaw Konstrukcji Maszyn Politechniki Śląskiej. Gliwice, 1991.
3. Wyczółkowski R.: "Zastosowanie Szkieletowego Systemu Doradczego W_SHELL do wspomagania procesu wyboru algorytmu", złożono do druku w materiałach IX konferencji "Polioptymalizacja i Komputerowe wspomaganie Projektowania" MIELNO 1991.

ANWENDUNG EINES EXPERTENSYSTEMS ZUR UNDERSTÜTZUNG DER AUSWAHL DER ALGORIYHMEN FÜR KONSTRUKTIONSBERECHNUNGEN

Zusammenfassung

Es wurde ein Expertensystem W_SHELL dargestellt, das im Lehrstuhl der Grundlagen des Maschinenbaus der TU Gliwice entwickelt wurde. Eine Möglichkeit der Ausnutzung dieses Expertensystems als eines Werkzeugs der Unterstützung der Berechnungen des Konstruktionsvorgangs wurde auch gezeigt. Um die im Referat angebotenen Lösung zu illustrieren, wurde ein Beispiel der Berechnung der grundsätzlichen Parametern eines Zahngetriebes ausgenutzt.

ZASTOSOWANIE SYSTEMÓW DORADCZYCH DLA POTRZEB WSPOMAGANIA WYBORU ALGORYTMU W OBLICZENIACH INŻYNIERSKICH

Streszczenie

W referacie przedstawiono szkieletowy system doradczy W_SHELL, opracowany w Katedrze Podstaw Konstrukcji Maszyn Politechnki Śląskiej, i pokazano możliwość wykorzystania tego systemu jako narzędzia wspomagania obliczeń w procesie projektowo-konstrukcyjnych. Dla zilustrowania zaproponowanego rozwiązania wykorzystano w referacie przykład obliczania podstawowych parametrów przekładni zębatej.

Wpłynęło do redakcji w styczniu 1992 r. Recenzent: Ewald Macha