INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED15
27-30 JULY 2015, POLITECNICO DI MILANO, ITALY

ASSEMBLY SEQUENCE PLANNING WITH THE PRINCIPLES OF DESIGN FOR ASSEMBLY

Sąsiadek, Michał
University of Zielona Góra, Poland

Abstract

The article characterises the EASYASSEMBLE method, which is meant to set an optimal assembly sequence for mechanical products, basing on an evaluation including requirements of design for assembly. The use of the computer implementation of the method - computer program EASYASSEMBLE - is presented on a real live example. The program is dedicated to constructors, technologists and planners (especially of the assembly process). It should serve as a help in designing products and adjusting their construction to easy and cheap assembly process.

Keywords: Design for X (DfX), Design engineering, Computer aided design (CAD)

Contact:
Dr.-Ing. Michał Sąsiadek
University of Zielona Góra
Mechanical Department
Poland
m.sasiadek@iizp.uz.zgora.pl

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

It is commonly known, that assembly process engages a good part of cost and time of the product manufacture and influences the product quality. Perhaps the best characterization has been formulated by Daniel E. Whitney (2004): “Assembly is the activity in which all the upstream processes of design, engineering, manufacturing, and logistics are brought together to create an object that performs a function”. Thus, every object of mechanical design should be easily and cheaply assembled. These requirements should be taken into account as early as possible during the design process. This is not easy because product specification includes often hundreds of contradictory items that need to be compromised.

In the traditional design process particular groups of requirements are allowed for one at a time, with those for assembly after others. Although having finished design document one can still make choice from a limited set of assembly processes, there would be, however, much better if the design object had not neglected assembly rules (Redford A.H., Chal J., 1994; Eder E.W., Hosnedl S., 2010). According to literature, the process constitutes about 30-50% of the cost of the product manufacturing (Anderson D.M., 2010; Boothroyd G, Dewhurst P., 1991; Jakubowski, J., Peterka, J., 2014). Labour intensity of the assembly process grows along with the number of product parts, and the proper conduct of this process depends on the order and the correct connection of the parts. Hence, it is important to determine the best sequence of connecting the components (assembly sequence) of the product. With a large number of components, the issue of determining the best assembly sequence is combinatorially complex (Sąsiadek M., Rohatyński R., 2008b). Determining the optimal (in accordance with the assumed requirements) assembly sequence without computer help is much more difficult with complex products.

Determining the optimal assembly sequence with simultaneous construction evaluation including the assembly requirements is significant, as noted above, especially at an early stage of the product design development, in its design conceptual phase. It is then possible to design the product concurrently with its assembly process, which greatly speeds up the product final implementation (Sąsiadek M. Rohatyński R., 2014c). Moreover, such an approach allows early identification of flaws in the construction (or its components) and necessary (from the assembly viewpoint) corrections at the design stage.

In the literature one can find a lot of views on the assembly sequence generating. Bourjault (1984) formulated an algorithm for generating all the permissible assembly sequences, which was based on a list of questions. These questions resulted in obtaining relations for the analysed constituents of a product. A similar algorithm is the one by De Fazio and Whitney (1987), however it is based on determining relations for assembly operations, which characterise pairs of combined parts. Sanderson and Homem de Mello (1991) developed an algorithm allowing to build a relational model, on the basis of which, using graph operations (graph cuts of and/or type), a set of all the possible assembly sequences was gained. Other studies related to determining the assembly sequences use for instance exploded views of the products, artificial intelligence methods. All the above approaches are applicable in the case of a previously developed product structure. Similarly, other approaches make the analysis of the assembly process possible, but only after the manufacturing stage, when the product components are ready and their assembly process is planned (Baldwin D.F. at al., 1991; Gottipolu R.B., Ghosh K., 1997; Qiang Su, 2007). In this case any construction changes are really expensive and involve redesign of the product and repeated production of components which have undergone construction changes.

Many of the authors, mentioned above, in order to verify assembly sequence take mostly under consideration specifications that result from actual assembly process. These are, time to complete the operation and a rates connected to handling and fitting. This type of verification does not allow for the possibility of construction change at the early level of design to modify for easier and cheaper assembly process in further phases of product development.

2 EASYASSEMBLE METHOD CHARACTERISTICS

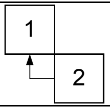
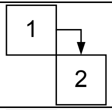
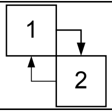
This proposed planning sequence of assembly method (called EASYASSEMBLE) takes under consideration overall rules of DFA to rate assembly connections. Thanks to that fact it can be used from an early stages of product development.

In proposed method four basic and implemented sequentially modules can be distinguished. In the first module the product design is mapped in the form of design structure matrix (M_k). In the second module, all relations stored in the M_k matrix are evaluated. The result of the implementation of the first two modules is a record of the product design structure in the form of M_k matrix and assigning an evaluation indicator to each relation stored in this matrix. In the third module precedence constraints for the assembly connections are defined (generated on the basis of the M_k matrix). They are taken into consideration to determine the set of feasible assembly sequences. The last module of the method is the algorithm for generating the set of feasible assembly sequences. The resulting assembly sequences are evaluated and recorded. It is possible to repeat the analysis with appropriate modifications in the individual steps of the method. The next section describes the method's modules.

2.1 Record of the product structure

All contact relations between the components of the product are identified on the basis of product design documentation. Contact relation is understood as the possibility of combining two parts. Established relations (connections) are stored in the form of a graph and the corresponding matrix - called further the relationship matrix or structure design matrix - M_k. This matrix has a size of $n\ x\ n$, where n is the number of the product components. Relations between the product components can assume three forms. The possible direction of the relations between the parts (direction of the assembly connection) are determined by the user (with the use of the programme) on the basis of the construction analysis, experience, and other aspects of the real assembly process. They are presented in the table 1.

Table 1 Forms of the matrix record of relations between the components

M_k matrix	Relation direction
$\begin{array}{ccc} & 1 & 2 \\ 1 & \blacksquare & \\ 2 & x & \blacksquare \end{array}$	⟹  ⟹ 1←2
$\begin{array}{ccc} & 1 & 2 \\ 1 & \blacksquare & x \\ 2 & & \blacksquare \end{array}$	⟹  ⟹ 2←1
$\begin{array}{ccc} & 1 & 2 \\ 1 & \blacksquare & x \\ 2 & x & \blacksquare \end{array}$	⟹  ⟹ 2←1 1←2

If there is no relationship between the parts (or if it is not possible connect two parts) no type of relation is assigned and the corresponding M_k matrix field stays empty.

2.2 Evaluation of the components connection

To evaluate a connection of two parts information from the literature (Booker J. D., Raines M., Swift K. G., 2001) was applied together with q_a indicator. It was developed on the basis of experts' knowledge and multiple analyses conducted in actual companies, and described in publications. The indicator was used to assess the set of connections defined earlier in the form of M_k matrix in order to evaluate assembly sequences and find the most propitious ones in the generated set. Moreover, it is assumed that it is going to be used to obtain information on the degree of complexity of the analysed structure and its component parts.

The possibility of defining values other than in the original study has been introduced. The values serve to evaluate particular components of the q_a indicator. This gives a chance to adjust the assessment with the use of q_a indicator to the specific conditions of a particular company, in which literature indicators would be wrongly applied for various reasons. In addition, the assessment value could be represented by cost or connection realisation time, which would facilitate defining of sequences characterised by the shortest time or the lowest realization cost. The components of $q_a=h_p \cdot f_p$ (Booker J. D., Raines M., Swift K. G., 2001) indicator:

- characteristics of passing and catching the component marked as h_p,
- characteristics of the process of combining the components f_p.

Indicator $f_p = A \cdot B \cdot C \cdot D \cdot E \cdot F \cdot G \cdot H$ takes into account:

- A - correctness of combining the parts in terms of the assembly function,

- *B* - necessity of precise mutual positioning of two parts to be combined,
- *C* - orienting of the parts to be combined,
- *D* - direction of combining the parts,
- *E* - type of connection, it depends on the contact surface between them,
- *F* - constraints of access and / or control of the connection,
- *G* - alignment and other possible obstacles,
- *H* - resistance of the parts combining.

Each integrant of the indicator f_p (*A* to *H*) is properly valued for each of the analysed component of the product according to the literature (Booker J. D., Raines M., Swift K. G., 2001).

2.3 Defining the constraints

Determination of the correct assembly sequences requires appropriate precedence constraints. They are related to the set of connections recorded in the M_k matrix. Each connection can be assigned to one of three designators:

- starting connections (p_s) - connections of two parts from which the creation of the assembly sequence variants of the product starts,
- connection 'skip' (p_p) - this connection is not taken into account when generating the assembly sequence variants of the product,
- blocking connection (p_b) - connection which prevents or limits getting a complete assembly in the later course of the assembly process.

The first type of constraints (starting connection) is used predominantly to define base components and parts from which the assembly sequence formation starts. Connections of the 'skip' type are defined in the case of reduction of a generated feasible assembly sequences set. This constraint can help to exclude resulting sequences with unfavourable sub-sequences.

The last of the constraints, and the most important one, is blocking connection, which has a direct influence on generating correct order of combining the parts, in terms of the selection completeness. This constraint is characteristic of those preceding connections, which prevent the realisation of the connection for which they are defined. This way the possibility of incorrect sequence when combining the parts is eliminated. It is assumed the blocking connections need to be defined with the operator 'and' ($\wedge$) and 'or' ($\vee$). In the first case, assigning the '$\wedge$' operator to the blocking connections ($p_{b1} \wedge p_{b2} \wedge \ldots \wedge p_{bn}$) means that connection p_n, for which the blocking connections are defined, can be executed before every blocking connection is made. Thus, it is possible to make *n-1* blocking connections before the connection p_n, for which *n* blocking connections were defined. If all the blocking connections are executed, it is impossible to achieve complete assembly of the whole product because realisation of the connection p_n is blocked. In the second case, assigning the '$\vee$' operator to the blocking connections ($p_{b1} \vee p_{b2} \vee \ldots \vee p_{bn}$) means that connection p_n, for which the blocking connections are defined, has to be executed before any of them. Even if one of the blocking connections is made, it is impossible to achieve complete assembly of the whole product because realisation of the connection p_n is blocked. Furthermore, it is possible to define blocking connection sequences (with the '$\wedge$' operator) separating them by the use of the '$\vee$' operator.

2.4 Algorithm for generating feasible sequences

The proposed algorithm for determining and evaluating the assembly sequences allows generation of all permissible variants for assembly sequences with simultaneous evaluation. It belongs to the category of classic (scientific) methods. In figure 1 the algorithm is presented in a schematic form, along with its particular steps. In the algorithm three databases have been distinguished. The first of them contains data related to the product structure and relations between its components. Directly from the database – 1 a list of possible connections is created. The first step of the algorithm is to choose the first available connection from the starting connections list and create an assembly subsequence from its components. At the same time, when selecting a starting connection, its evaluation from the database - 2 is taken. This database contains information pertaining to evaluation of all the relations between the connections' components. This subsequence is recorded as the M_{K+1} matrix, which decreases the size of the M_K matrix by 1. Relations recorded in the M_K matrix are changed into the form of the M_{K+1} matrix and constraints for the current subsequence are checked. All the constraints (connections of 'skip' type, blocking connections of 'OR' and 'AND' type) are recorded in the database 3. If there are any constraints, the current sequence is excluded from further consideration. If the constraints allow continuous building of the assembly sequence, more components are added. Subsequences of a higher order are created until a complete sequence meeting all the constraints is built. Produced sequences are then recorded and the starting connection used in the process is deleted from the list of available connections. Next, the algorithm chooses another available starting connection and the process of sequences creation is repeated. After every starting connection is used a set of all the possible assembly sequences is received.

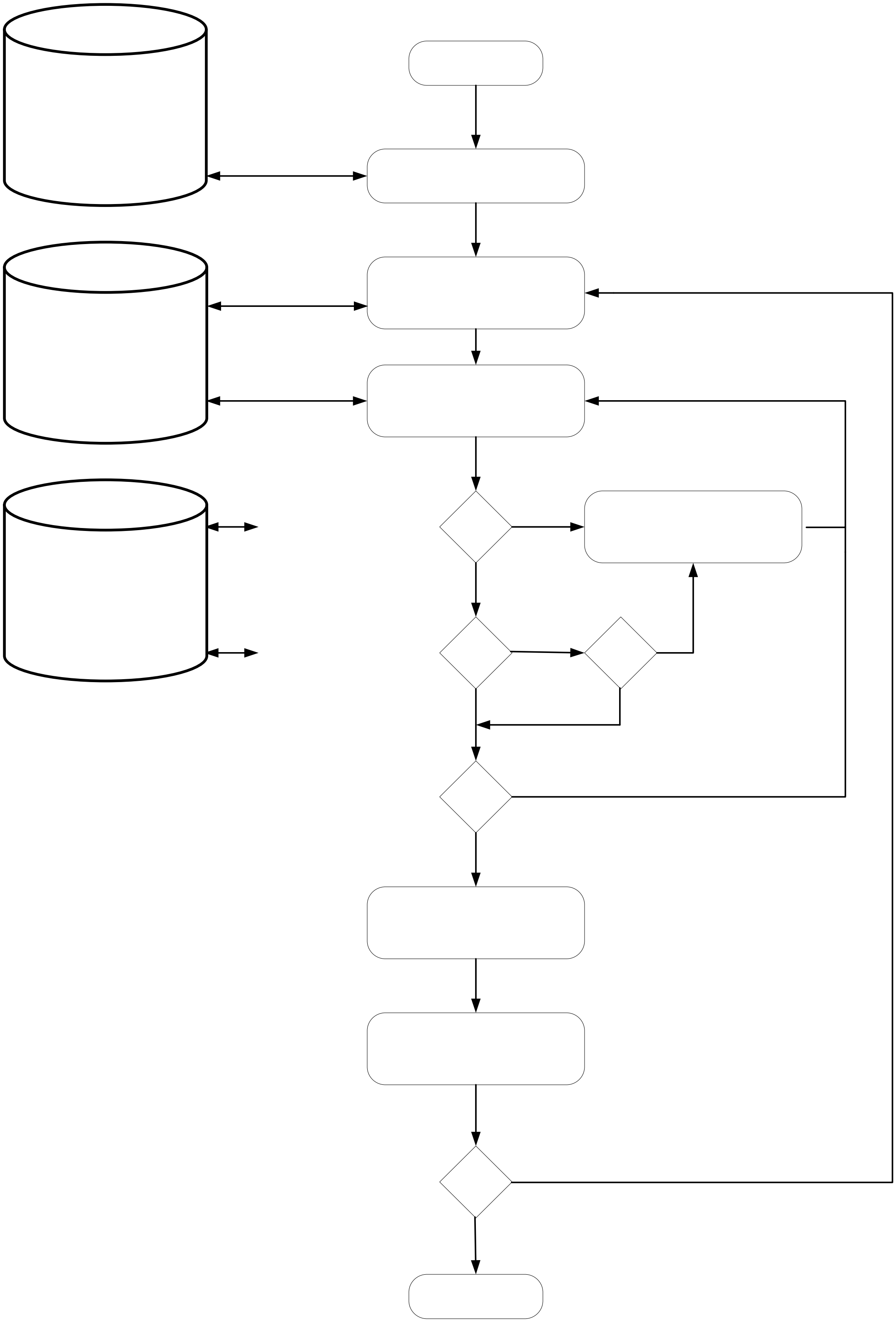
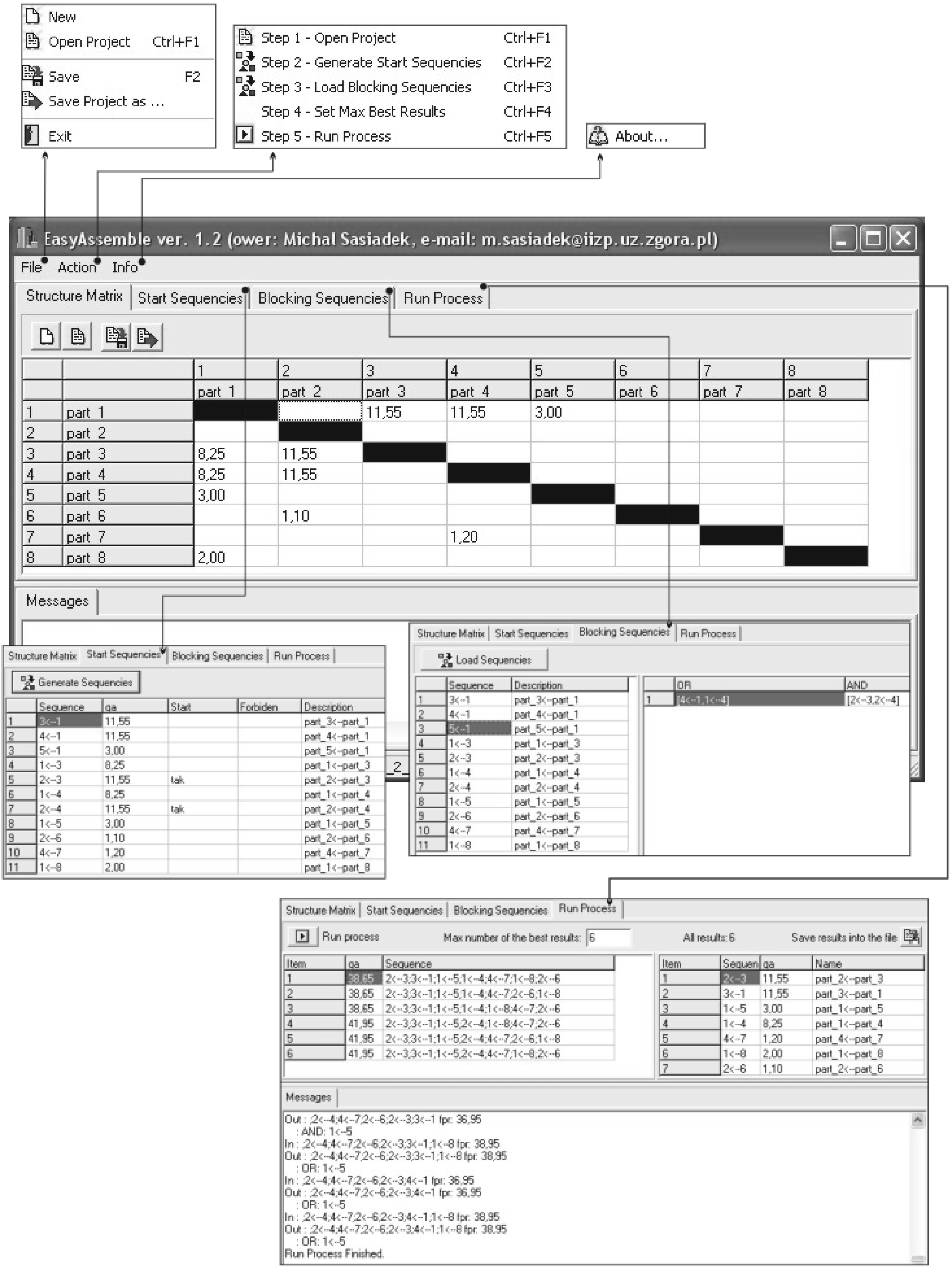


Figure 1 Algorithm for generating feasible assembly sequences

3 COMPUTER IMPLEMENTATION OF THE METHOD

The result of computer implementation of the method is EASYASSEMBLE programme. Four tabs of the programme are presented in the figure 1. In the first tab, *Structure Matrix*, the user defines relations between the parts and assigns their constituent values (h_p, f_p) of the grade indicator q_a (Booker J. D., Raines M., Swift K. G., 2001). In the next tab, *Start Sequences*, the programme generates the set of allowable operations out of which the user has the possibility of selecting the operations of "start" and "ignore" type. In the *Blocking Sequences* tab, there are limitations of "OR" and/or "AND" type. All the information defined in the first three tabs are saved in a file with *.asp filename extension (abbreviation of assembly sequence planning).



		1	2	3	4	5	6	7	8
		part 1	part 2	part 3	part 4	part 5	part 6	part 7	part 8
1	part 1	■		11,55	11,55	3,00			
2	part 2		■						
3	part 3	8,25	11,55	■					
4	part 4	8,25	11,55		■				
5	part 5	3,00				■			
6	part 6		1,10				■		
7	part 7				1,20			■	
8	part 8	2,00							■

	Sequence	qa	Start	Forbiden	Description
1	3<-1	11,55			part_3<-part_1
2	4<-1	11,55			part_4<-part_1
3	5<-1	3,00			part_5<-part_1
4	1<-3	8,25			part_1<-part_3
5	2<-3	11,55	tak		part_2<-part_3
6	1<-4	8,25			part_1<-part_4
7	2<-4	11,55	tak		part_2<-part_4
8	1<-5	3,00			part_1<-part_5
9	2<-6	1,10			part_2<-part_6
10	4<-7	1,20			part_4<-part_7
11	1<-8	2,00			part_1<-part_8

	Sequence	Description
1	3<-1	part_3<-part_1
2	4<-1	part_4<-part_1
3	5<-1	part_5<-part_1
4	1<-3	part_1<-part_3
5	2<-3	part_2<-part_3
6	1<-4	part_1<-part_4
7	2<-4	part_2<-part_4
8	1<-5	part_1<-part_5
9	2<-6	part_2<-part_6
10	4<-7	part_4<-part_7
11	1<-8	part_1<-part_8

	OR	AND
1	[4<-1,1<-4]	[2<-3,2<-4]

Max number of the best results: 6

Item	qa	Sequence
1	38,65	2<-3;3<-1;1<-5;1<-4;4<-7;1<-8;2<-6
2	38,65	2<-3;3<-1;1<-5;1<-4;4<-7;2<-6;1<-8
3	38,65	2<-3;3<-1;1<-5;1<-4;1<-8;4<-7;2<-6
4	41,95	2<-3;3<-1;1<-5;2<-4;1<-8;4<-7;2<-6
5	41,95	2<-3;3<-1;1<-5;2<-4;4<-7;2<-6;1<-8
6	41,95	2<-3;3<-1;1<-5;2<-4;4<-7;1<-8;2<-6

Item	Sequen	qa	Name
1	2<-3	11,55	part_2<-part_3
2	3<-1	11,55	part_3<-part_1
3	1<-5	3,00	part_1<-part_5
4	1<-4	8,25	part_1<-part_4
5	4<-7	1,20	part_4<-part_7
6	1<-8	2,00	part_1<-part_8
7	2<-6	1,10	part_2<-part_6

```
Out : ,2<-4;4<-7,2<-6;2<-3;3<-1 fpr: 36,95
   : AND: 1<-5
In : ,2<-4;4<-7,2<-6;2<-3;3<-1;1<-8 fpr: 38,95
Out : ,2<-4;4<-7,2<-6;2<-3;3<-1;1<-8 fpr: 38,95
   : OR: 1<-5
In : ,2<-4;4<-7,2<-6;2<-3;4<-1 fpr: 36,95
Out : ,2<-4;4<-7,2<-6;2<-3;4<-1 fpr: 36,95
   : OR: 1<-5
In : ,2<-4;4<-7,2<-6;2<-3;4<-1;1<-8 fpr: 38,95
Out : ,2<-4;4<-7,2<-6;2<-3;4<-1;1<-8 fpr: 38,95
   : OR: 1<-5
Run Process Finished.
```

Figure 2 EASYASSEMBLE dialog boxes

In the last tab, *Run Process*, an algorithm generating allowable assembly sequence according to previously defined *.asp file is performed. The user has the possibility of reviewing the results and saving them to a text file (*.txt) as well as to obtain the information concerning particular steps of the algorithm. Figure 2 presents selected dialog boxes of EASYASSEMBLE programme. Previously characterised tabs are presented in the form of dialog boxes. This program has been described in detail by Sąsiadek M. (2013a).

4 METHOD AND EASYASSEMBLE PROGRAMME APPLICATION

Application of the programme is presented based on the set of assembly sequences determined for a piston-rod unit. Piston-rod unit, presented in the figure 3, consists of 11 constituent parts: 1 – Piston-rod, 2 – Muffler front sleeve, 3 – Muffler rear sleeve, 4 – Front gland ring, 5 – Rear gland ring, 6 – Piston, 7.1 – U1 sealing, 7.2 – U1 sealing, 8 – M4x10 screw, 9 – Tab washer, 10 – Piston sealing.

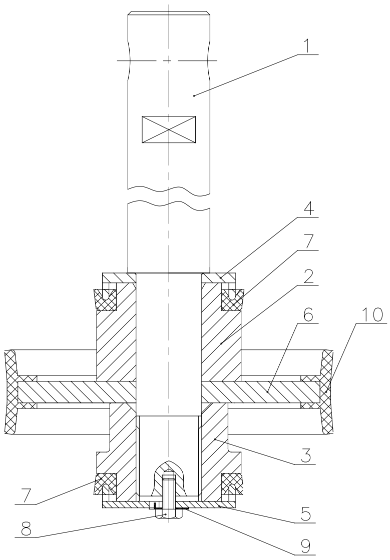


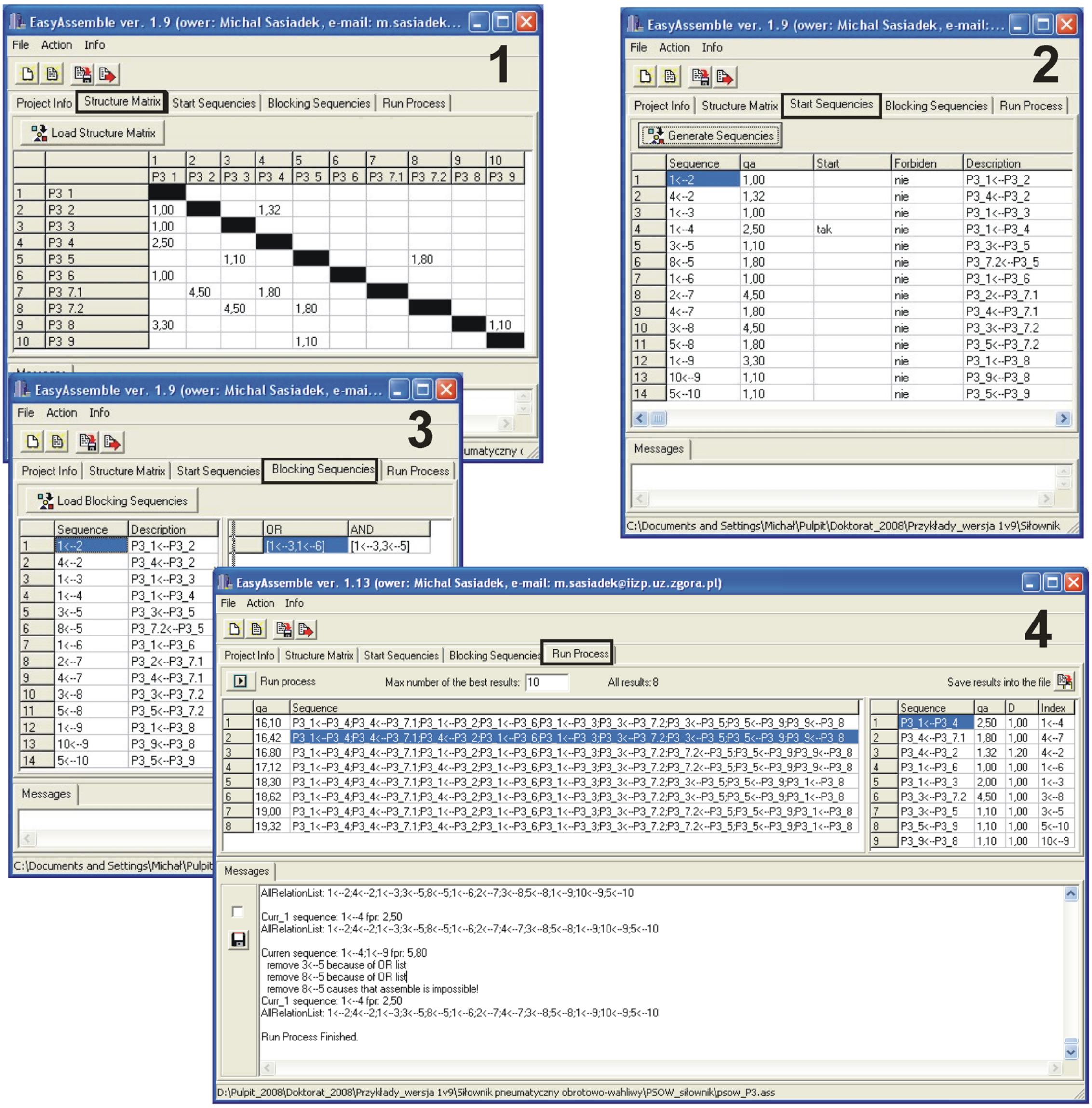
Figure 3 Piston-rod unit

In table 2, 14 assembly connections are collated. One of them is marked as a starting connection (fitting of part 4 to part 1, 1←4). Additionally, to generate proper sequences, blocking connections of type ‘AND’ and ‘OR’ are defined for corresponding assembly connections, which are also present on the chart. Among the constituents of the piston-rod unit one component was distinguished. It consists of a piston and its gasket (parts 6 and 10) and it is marked in the tables as 6-10 (in the programme its identifier is P3_6).

Table 2. Connections and constraints of the piston-rod unit

No.	Assembly connection	Starting connection	Blocking connections	
			„OR” ; ∨	„AND” ; ∧
1	1←2	--	1←6-10 ∨ 1←3	1←3 ∧ 3←5
2	1←3	--	--	--
3	1←4	x	--	--
4	1←6-10	--	1←3	1←3 ∧ 3←5
5	1←8	--	--	--
6	2←7.1	--	4←2 ∨ 1←2	--
7	3←7.2	--	3←5	--
8	3←5	--	1←8	--
9	4←7.1	--	4←2 ∨ 1←2	--
10	4←2	--	1←6-10 ∨ 1←3	1←3 ∧ 3←5
11	5←7.2	--	3←5	--
12	5←9	--	1←8	--
13	7.2←5	--	1←8	--
14	9←8	--	--	--

Each of the assembly connections has been rated according to the indicator qa which characterises its capability of easy and cheap assembly. In figure 4 four programme tabs are presented. They are used to conduct consecutive steps for determining a set of permissible (correct) assembly sequences and choosing the best of them. The dialog box 1 displays a record of the product design structure. It depicts all defined assembly connections and the indicator q_a value attributed to them. 14 connections stored in a structure matrix have been determined on the basis of the component structure analysis. The dialog box 2 displays a list of available assembly sequences with distinguished connections 'start' and 'forbidden'. The dialog box 3 displays a constraint for all particular assembly connections in the form of connections 'AND' and/or 'OR'. All the information in the program is in accordance with the specification shown in table 2. The last dialog box (number 4) is used to display results. The algorithm operation resulted in receiving 8 assembly sequences seen in this box. Q_a value for each of the generated sequence is calculated as the sum of the partial q_a values of each connection present in the sequence.



		1	2	3	4	5	6	7	8	9	10
		P3 1	P3 2	P3 3	P3 4	P3 5	P3 6	P3 7.1	P3 7.2	P3 8	P3 9
1	P3 1	■									
2	P3 2	1,00	■		1,32						
3	P3 3	1,00		■							
4	P3 4	2,50			■						
5	P3 5			1,10		■			1,80		
6	P3 6	1,00					■				
7	P3 7.1		4,50		1,80			■			
8	P3 7.2			4,50		1,80			■		
9	P3 8	3,30								■	1,10
10	P3 9					1,10					■

umatyczny (

	Sequence	qa	Start	Forbiden	Description
1	1<--2	1,00		nie	P3_1<--P3_2
2	4<--2	1,32		nie	P3_4<--P3_2
3	1<--3	1,00		nie	P3_1<--P3_3
4	1<--4	2,50	tak	nie	P3_1<--P3_4
5	3<--5	1,10		nie	P3_3<--P3_5
6	8<--5	1,80		nie	P3_7.2<--P3_5
7	1<--6	1,00		nie	P3_1<--P3_6
8	2<--7	4,50		nie	P3_2<--P3_7.1
9	4<--7	1,80		nie	P3_4<--P3_7.1
10	3<--8	4,50		nie	P3_3<--P3_7.2
11	5<--8	1,80		nie	P3_5<--P3_7.2
12	1<--9	3,30		nie	P3_1<--P3_8
13	10<--9	1,10		nie	P3_9<--P3_8
14	5<--10	1,10		nie	P3_5<--P3_9

C:\Documents and Settings\Michał\Pulpit\Doktorat_2008\Przykłady_wersja 1v9\Siłownik

	Sequence	Description
1	1<--2	P3_1<--P3_2
2	4<--2	P3_4<--P3_2
3	1<--3	P3_1<--P3_3
4	1<--4	P3_1<--P3_4
5	3<--5	P3_3<--P3_5
6	8<--5	P3_7.2<--P3_5
7	1<--6	P3_1<--P3_6
8	2<--7	P3_2<--P3_7.1
9	4<--7	P3_4<--P3_7.1
10	3<--8	P3_3<--P3_7.2
11	5<--8	P3_5<--P3_7.2
12	1<--9	P3_1<--P3_8
13	10<--9	P3_9<--P3_8
14	5<--10	P3_5<--P3_9

	OR	AND
	[1<--3,1<--6]	[1<--3,3<--5]

C:\Documents and Settings\Michał\Pulpit

Max number of the best results: 10

All results: 8

	qa	Sequence
1	16,10	P3_1<--P3_4;P3_4<--P3_7.1;P3_1<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_3<--P3_5;P3_5<--P3_9;P3_9<--P3_8
2	16,42	P3_1<--P3_4;P3_4<--P3_7.1;P3_4<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_3<--P3_5;P3_5<--P3_9;P3_9<--P3_8
3	16,80	P3_1<--P3_4;P3_4<--P3_7.1;P3_1<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_7.2<--P3_5;P3_5<--P3_9;P3_9<--P3_8
4	17,12	P3_1<--P3_4;P3_4<--P3_7.1;P3_4<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_7.2<--P3_5;P3_5<--P3_9;P3_9<--P3_8
5	18,30	P3_1<--P3_4;P3_4<--P3_7.1;P3_1<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_3<--P3_5;P3_5<--P3_9;P3_1<--P3_8
6	18,62	P3_1<--P3_4;P3_4<--P3_7.1;P3_4<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_3<--P3_5;P3_5<--P3_9;P3_1<--P3_8
7	19,00	P3_1<--P3_4;P3_4<--P3_7.1;P3_1<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_7.2<--P3_5;P3_5<--P3_9;P3_1<--P3_8
8	19,32	P3_1<--P3_4;P3_4<--P3_7.1;P3_4<--P3_2;P3_1<--P3_6;P3_1<--P3_3;P3_3<--P3_7.2;P3_7.2<--P3_5;P3_5<--P3_9;P3_1<--P3_8

	Sequence	qa	D	Index
1	P3_1<--P3_4	2,50	1,00	1<--4
2	P3_4<--P3_7.1	1,80	1,00	4<--7
3	P3_4<--P3_2	1,32	1,20	4<--2
4	P3_1<--P3_6	1,00	1,00	1<--6
5	P3_1<--P3_3	2,00	1,00	1<--3
6	P3_3<--P3_7.2	4,50	1,00	3<--8
7	P3_3<--P3_5	1,10	1,00	3<--5
8	P3_5<--P3_9	1,10	1,00	5<--10
9	P3_9<--P3_8	1,10	1,00	10<--9

```
AllRelationList: 1<--2;4<--2;1<--3;3<--5;8<--5;1<--6;2<--7;3<--8;5<--8;1<--9;10<--9;5<--10

Curr_1 sequence: 1<--4 fpr: 2,50
AllRelationList: 1<--2;4<--2;1<--3;3<--5;8<--5;1<--6;2<--7;4<--7;3<--8;5<--8;1<--9;10<--9;5<--10

Curren sequence: 1<--4;1<--9 fpr: 5,80
  remove 3<--5 because of OR list
  remove 8<--5 because of OR list
  remove 8<--5 causes that assemble is impossible!
Curr_1 sequence: 1<--4 fpr: 2,50
AllRelationList: 1<--2;4<--2;1<--3;3<--5;8<--5;1<--6;2<--7;4<--7;3<--8;5<--8;1<--9;10<--9;5<--10

Run Process Finished.
```

D:\Pulpit_2008\Doktorat_2008\Przykłady_wersja 1v9\Siłownik pneumatyczny obrotowo-wahliwy\PSOW_siłownik\psow_P3.ass

Figure 4 Result dialog boxes of the servo-motor piston-rod unit

5 SUMMARY AND CONCLUSION

This paper describes a new, computer aided method for planning the best assembly sequence of a designed product components in terms of criteria for design for assembly. A computer programme is also presented and used on a simple example. The proposed method can be efficiently applied to find the most advantageous assembly sequence for the constituent parts of a unit or machine. It can also be used to modify constructions in order to increase the installation efficiency.

The algorithm for determining the assembly sequence implemented in the programme is different from former approaches, mentioned in the first part of the article. First of all, it can be used at an early stage of the product design (conceptual phase), when the designer has plenty of options to modify the structure without increasing the costs and extending the realisation time of the designed product. The data needed to the analysis is the knowledge of the structure of the analysed product and its components as well as the types of connections between these components. Moreover, the used assembly connections' evaluation, allowing for the rules of design for assembly, contributes to development of constructions adjusted to a simple and fast assembly process.

The developed computer programme can be used to determine a set of feasible assembly sequence for products of various types. The q_a indicator used to evaluate the assembly connections may allow for manual or automatic assembly process. In the case of complex products, consisting of more than 40 elements, it is beneficial to conduct a preliminary product decomposition and define sub-assemblies constituting subtasks. This operation is useful in terms of decreasing the complexity of a task. In such a situation an overall solution is achieved via aggregation of partial solutions.

It is assumed that the further author's work is going to be focused on increasing the efficiency of applying a computer programme for complex products by:

- possibility of defining the structure of sub-assemblies in a single file engaging the whole product,
- implementation of an additional computational module based on artificial intelligence (genetic or evolutionary algorithm) in the programme algorithm,
- development of a knowledge base containing information on the correct design solutions of two combined components of a product and coupling this base with a requesting module, whose task will be to summarise information (advice) related to improvement in the construction of individual constituent parts in order to increase the efficiency of the combining process,
- integration of the developed method with the CAD system, to automatically generate contact relations and possible connections between the elements of the designed construction.

REFERENCES

Anderson D.M. (2010) Design for Manufacturability & Concurrent Engineering, CIM Press.

Baldwin D.F., Abell TE, Lui MC, de Fazio TL, Whitney DE (1991). An integrated computer aid for generating and evaluating assembly sequences for mechanical products. The International Journal of Advanced Manufacturing Technology, Vol.7, No.1, pp. 78–89.

Booker J. D., Raines M., Swift K. G. (2001) Designing Capable and Reliable Products, Butterworth-Heinemann.

Boothroyd G, Dewhurst P. (1991) Design for Assembly, Boothroyd Dewhurst Inc., Wakefield.

Bourjault A. (1984) Contribution à une approche méthodologique del'assemblage automisé: elaboration automatique des sequences opératoires. France.

De Fazio T.L., Whitney D.E. (1987) Simplified generation of all mechanical assembly sequences. IEEE Journal of Robotics and Automation, Vol. 3, No. 6, pp. 640–656.

Eder E.W., Hosnedl S. (2010) Introduction to design engineering. Systematic creativity and management, Taylor &Francis Group, London, UK.

Gottipolu R.B., Ghosh K. (1997) Representation and selection of assembly sequences in computer-aided assembly process planning, International Journal of Production Research, Vol. 35, No. 12, pp. 3447-3465.

Jakubowski, J., Peterka, J. (2014) Design for manufacturability in virtual environment using knowledge engineering, Management and Production Engineering Review, Vol. 5, No. 1, pp. 3-10.

Qiang Su (2007) Computer aided geometric feasible assembly sequence planning and optimizing, The International Journal of Advanced Manufacturing Technology, Vol. 33, pp. 48–57.

Redford A.H., Chal J. (1994), Design for Assembly: Principles and Practice, McGraw-Hill.

Sanderson A.C., Homem de Mello L.S., Zhang H. (1990) Assembly sequence planning. AI Mag, Vol. 11, No. 1, pp. 62–81.

Sąsiadek M. (2013a) Computer aided mechanical assembly sequence planning, 7th International Conference on Research and Practical Issues of Enterprise Information Systems - CONFENIS 2013, Prague, Czech Republic, Trauner Verlag, pp. 335-343.

Sąsiadek M. Rohatyński R. (2008b) A method of computer aided design for assembly, Proceedings of the Design methods for industrial practice Conference, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, pp. 119-126.

Sąsiadek M. Rohatyński R. (2014c) Design for disassembly and reverse logistics - a new challenge for concurrent engineering, Tools and Methods of Competitive Engineering, Budapest, Hungary, pp. 1627-1634.

Whitney D.E. (2004) Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development, Oxford University Press.