INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

INFORMATION IMPORTANCE AND DESCRIPTOR INSTANTIATION EFFORT IN ENGINEERING DESIGN

Argiris J. Dentsoras[1]
[1]Associate Prof., Dept. of Mechanical Engineering and Aeronautics, University of Patras, Greece

ABSTRACT
The paper studies the process of information generation during design and introduces new metrics regarding: a. the quantitative determination of its importance and b. the effort required for the instantiation of the design entities representing the design problem.
Binary relationships among the design entities (descriptors) form tree structures that represent the objective associative design knowledge. The distinct roles that the two categories of these descriptors, namely dependent and primary, play in the design process are discussed and additional factors that define realistically the required effort for the instantiation of every primary descriptor are introduced.
The binary relationships among the descriptors form a dependency matrix, which is further transformed through a number of operations based on the connectivity among the descriptors and the aforementioned additional factors. From the resulting final matrix, a unique sorted list of primary design descriptors is produced. This list may be used for the sequential instantiation of these descriptors so that production of design information of maximum importance and least effort takes place during the early design stages and under completely realistic design conditions. A short design example contributes towards a better understanding of the proposed approach.

Keywords: Design, information importance, design descriptors, instantiation effort

1 INTRODUCTION
Engineering design is an activity of intensive manipulation of knowledge. The goal in design is always the generation of design solutions that satisfy the functional requirements formally stated as engineering measurable quantities. Being an activity of knowledge representation and manipulation, design has been the subject of extended research during the last decades. In combination with the accumulated design experience and the advances achieved in other scientific fields such as Artificial Intelligence, Computer Graphics, Soft Computing, etc., significant progress has been made towards the establishment of reliable models for the representation and handling of the design knowledge.
In every design problem, there is always the need to establish associative relationships (functional, topological etc.) among the different design entities. These relationships are mainly hierarchical, form a major part of the design knowledge and are, by all means, necessary for the guidance of the instantiation process for these entities throughout the design process [1], [2].
Design data are used for initiating, maintaining and finally completing the design process. Part of this data play a decisive role from the very beginning of the design process and affect its evolution. They are usually involved in many design tasks, cover multiple design stages and affect value assignment procedures and decision-making processes. Some other data are less important and their impact is restricted in small design subspaces, usually treated during the later design phases.
Concerning the information handled and/or produced during design, S. B. Shooter et al [3] present a model for the flow of design information that is sufficiently formal in order to support a semantics-based approach for developing information exchange standards. The model classifies design information into various types, organizes these types into information states and levels of abstraction and identifies the various transformations that operate within and between the information states. Lee

et al [4] extend the use of entities in order to represent both the design information and the activity involved in design.

In an earlier paper by Warfield [5], the use of binary matrices in system modeling and the development of such matrices for the representation of structural models of systems had been discussed. Later, Steward [6] introduced the Design Structure Matrix (DSM) as a method for managing the design of complex systems. The proposed method handles design iterations and reviews, shows the information flow and determines the consequences of a change in any variable on the rest of the variables. Kusiak et al [7], [8], [9] used DSM within the framework of concurrent engineering for the decomposition and scheduling of design activities, Eppinger [10] presented model-based approaches for managing concurrent engineering, Black et al [11] used DSM for the design of an automotive break system and Austin et al [12] used the same approach for the manipulation of the flow of design information in building design.

A. Yassine et al [13] proposed a qualitative approach to engineering design management from an information structure perspective and enhanced the classical design structure matrix method by introducing the notion of structural sensitivity analysis. Finally, Browning [14] – in a review paper - discussed the advantages of DSM vs. alternative system representation and analysis techniques and posed new research directions.

H. A Bashir and V. Thomson., in a series of papers, study – among others – design effort. In [15] they consider design projects overruns and identify underestimation of design effort as one of the major reasons for the problem. They also suggest adoption of a metrics approach to estimate design effort as a promising way of overcoming this difficulty. In [16] they propose parametric estimation models based on product functional decomposition, whereas in [17] they describe an analogy-based model for estimating design effort. Finally, K. Hölttä and K. N. Otto [18] introduce a design based on modularity methods and a novel redesign effort complexity metric that helps define module boundaries so that changes in the modules require minimum redesign effort.

Dentsoras [19] and Tsalidis and Dentsoras [20] proposed a representation formalism based on a separation of design entities into a set of dependent and a set of non-dependent (primary) entities. The data for the instantiation of the non-dependent entities had been considered as provided by the designer and/or are extracted from other sources (databases, previous design cases etc.).

In accordance with the work by Dentsoras [19] and Tsalidis and Dentsoras [20], Drakatou and Dentsoras [1] investigated the possibility of early instantiations of design entities that present a major importance in the design process. They assumed that the design knowledge configuration could provide a set of non-dependent entities that may be further sorted according to their occurrence in this configuration. The relation between this configuration and the corresponding set of non-dependent design entities was also studied and it was shown that a unique list of non-dependent entities may be derived ensuring the derivation of the maximum number of instantiations of dependent entities during the very beginning of the design process.

In a later paper by Dentsoras [2], the characteristics of the dependent and the primary entities were further studied and their distinct roles in the design process were discussed. New terms (design information importance, required design effort, design cycle, importance – effort curves, etc.) were introduced and a simpler treatment of the design knowledge through matrix manipulations and rearrangements was presented. A unique sorted instantiation list of primary entities was suggested that ensured the production of design information of maximum importance with the least effort in the early design stages.

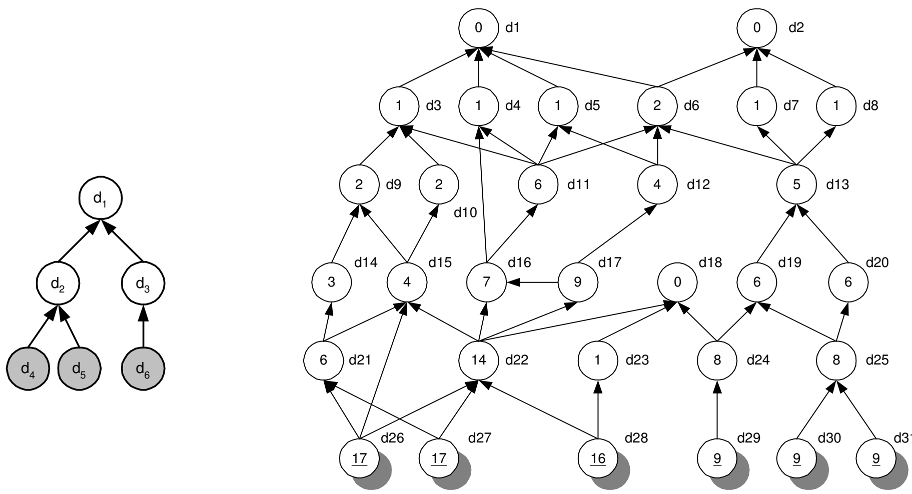
In the present paper, the work done in [2] is further extended towards the direction of studying any possible diversity that may exist for the amount of the required instantiation effort among the different design entities. By taking into account additional factors for the primary design entities, the proposed method is modified so that, after a series of proper matrix manipulations and rearrangements, the produced list conforms better to the practice followed for the instantiation of the design entities in real world design problems.

The objectives set for the present work are: a. the suggestion of a set of factors for the representation of various features of the design entities that cannot be expressed via the simple binary associative relationships used in [2] and b. the formation of a new instantiation list which, based on these factors, could reflect more realistically the design practice followed in solving real-world design problems.

2 EFFORT, IMPORTANCE AND OCCURRENCE – AN INTEGRATED APPROACH

2.1 Basic term definitions

The term *design descriptor* is used in order to represent any *meaningful, identifiable and unique word or phrase that describes a design entity*. A *primary descriptor* is *a descriptor that possesses the unique property to be input (primary, key) object* for the design problem under consideration. Once instantiated, the primary descriptor determines, through the established associative relationships, the rest, *non-primary, dependent descriptors*. In Figure 1 associative relationships among descriptors are shown in the form of two digraphs. In Figure 1.a, the grey-shaded descriptors d_4, d_5 and d_6 are primary and the rest are dependent.



a. b.

Figure 1. Associative relationships among descriptors: a. Primary and dependent descriptors, b. Importance values of descriptors in a digraph.

The *importance* (or *importance value*) of a dependent design descriptor is defined as *the total number of other, dependent descriptors whose instantiation depends upon it*. In the digraph of Figure 1.b in every node – except for the nodes representing primary descriptors - the corresponding importance value is given. The underlined value in a node that represents a primary descriptor in Figure 1.b corresponds to the *number of dependent descriptors whose instantiation depends upon this primary descriptor*. This number represents the *occurrence* of the primary descriptor.

The *instantiation of a primary descriptor is an event* that takes place during the design process and *constitutes a design cycle.* In *every design cycle* it is assumed that *a recursive value propagation process is "fired" so that the maximum possible – for that cycle - dependent descriptor instantiations take place.* The *sum of the instantiation efforts for all primary descriptors that are necessary for the instantiation of a dependent descriptor* is defined as *the design effort* for this descriptor. The *design process* is considered as *completed* if *all (dependent and non-dependent) descriptors have been instantiated.*

2.2 Factors that affect design effort

In [2] it was assumed that the required instantiation effort – formally calculated - was the same for every primary descriptor. This assumption facilitated the establishment of a formal method for calculating the importance of the generated information during the design process. What is more important, the method introduced the idea of a sorted list of primary descriptors that could be used as a guide for the designer in order to obtain – during the descriptor instantiation process - maximum design information with the least effort. The example case studied in [2] and, since then, some additional design cases subjected to a similar treatment showed the conformation of the proposed method with the established practice for the majority of the examined cases.

In the present work, the work done in [2] is further extended towards the direction of studying any possible diversity that may exist for the amount of the required instantiation effort among the different primary descriptors. The main reason for this extension is to reflect in a more realistic way the current practice regarding the design process where such a differentiation is observed for the majority of the cases. Indeed, it seems reasonable to claim, for example, that more effort is required to assign values to a set of primary descriptors during innovative design than during routine and/or parametric design. For the first case, there isn't adequate experience and knowledge and this fact implies that more resources and time and will be necessary in order to instantiate these descriptors and vice versa. Furthermore, the determination of some primary descriptors in a design problem may constitute a cumbersome process if, for example, extensive runs of repetitive arithmetic procedures are needed while some other descriptors would need – for their instantiation- only some milliseconds of access time of a database.

In [16], K. Hölttä and K. N. Otto reviewed previously published research results and presented some major factors that contribute for the estimation of the required effort in a design project. These factors are: a. product complexity, b. severity of requirements, c. use of new technology, d. experience and skill, e. design team size and methods of communication, f. use of design assisted tools and g. use of a formal process.

In the present work the factors introduced in [16] are taken into account in order to introduce six (6) new independent generic factors that differentiate the required instantiation effort among the different primary descriptors in a design problem. Every such factor maps the intensity of a certain general characteristic that a primary descriptor may present to a certain value within a standard arithmetic scale for every primary descriptor. These factors may affect all descriptors or just a subset of them. Additionally, their effect may be the same for all the affected descriptors or it may be differentiated for every different set of descriptors or for each individual descriptor. The factors are:

- The *awareness of the designer and/or the design team* – as a result of expertise and scientific knowledge - *regarding the procedure of determining the value of a descriptor*. The more familiar and the more aware is a designer or a team the less is the effort needed for the instantiation of a primary descriptor
- The *objective complexity of the procedure* needed *for the determination of the value of a descriptor*. The more complex the procedure the more effort is needed for its instantiation
- The *constraints* that restrict the value range for the descriptor. It is probable that a wide range of permissible values will provoke a more complicated and thus more "cumbersome" process
- The *standardization* that restricts the values the descriptor. It is probable that the absence of standardization will also provoke a more complicated instantiation process
- The *amount of collaborative work needed*. If a designer in a design team has to make extensive collaborative work in order to assess the value of primary descriptor this will increase the overall required effort
- The *amount of resources* – other that human – needed (software, hardware, etc)

The above factors should be taken into account for each primary descriptor in order to determine the required effort for its instantiation. If $r_{da}, r_{oc}, r_{cs}, r_{st}, r_{cw}, r_{ra}$ are the symbols for each one of them (the symbols correspond to the order of appearance of each factor in the above list), then each factor may get values from a standard arithmetic scale. For example, if the factor r_{da} for the awareness of the designer regarding the design problem is considered, then it could vary in the range 1 (fully aware) to 10 (absolutely unaware).

2.3 Calculations of effort, importance and occurrence

Consider a design problem with n descriptors that form a set D with $|D| = n, n \geq 2$. There are m dependent descriptors forming a subset S of set D and p primary descriptors forming a subset T of the same set. If s_k is a dependent descriptor in S, then there is a subset T_k of primary descriptors in T whose instantiations affect the instantiation of s_k (see figure 1.b). The *design effort index* $E(s_k)$ for s_k is defined as:

$$E\left(s_k\right)=\sum_{j=1}^{|T_k|}(r_{sum})_j, k=1,2,...,m \tag{1}$$

where $(r_{sum})_j$ is the sum of factors $r_{da}, r_{oc}, r_{cs}, r_{st}, r_{cw}, r_{ra}$ for the j- descriptor of T_k and $|T_k|$ is the cardinality of set T_k.

The design effort indices of all dependent descriptors in S form a *design effort vector* E and $b_{\max}=\max\left(E\left(s_1\right), E\left(s_2\right),..., E\left(s_m\right)\right)$, then the *normalized design effort vector* is defined as:

$$E_n=\left\{E\left(s_1\right)/b_{\max}, E\left(s_2\right)/b_{\max},..., E\left(s_m\right)/b_{\max}\right\} \tag{2}$$

If s_i is a dependent descriptor in S, then there is a subset S_i of descriptors other than s_i in S, whose instantiations are affected by s_i. The *importance index* $I(s_i)$ of s_i is defined as:

$$I\left(s_i\right)=\left|S_i\right|\geq 0 \ \ with \ \ I\left(s_i\right)\leq\left(m-1\right) \tag{3}$$

where $|S_i|$ is the cardinality of set S_i.The importance indices of all descriptors in S form an *importance vector* I. If the maximum index value is $a_{\max}=\max(I(s_1), I(s_2),..., I(s_m))$, then the *normalized importance vector* is defined as:

$$I_n=\left\{I\left(s_1\right)/a_{\max}, I\left(s_2\right)/a_{\max},..., I\left(s_m\right)/a_{\max}\right\} \tag{4}$$

If t_j is a primary descriptor in T, then there is a subset S_j of descriptors in S whose instantiation depends upon t_j. Then the *occurrence index* $O(t_j)$ of descriptor t_j is defined as:

$$O\left(t_j\right)=\left|S_j\right|\geq 1 \ \ with \ \ O\left(S_j\right)\leq m. \tag{5}$$

where $|S_j|$ is the cardinality of set S_j.

The occurrence indices of all descriptors in T form an occurrence vector O. If the maximum index value in this vector is $c_{\max}=\max\left(O\left(t_1\right), O\left(t_2\right),..., O\left(t_p\right)\right)$, then the *normalized occurrence vector* is defined as:

$$O_n=\left\{O\left(t_1\right)/c_{\max}, O\left(t_2\right)/c_{\max},..., O\left(t_p\right)/c_{\max}\right\} \tag{6}$$

2.4 The dependency matrix

The sets S and T can be used in order to form a $(|S|+1)\times(|T|+1)$ *dependency matrix* whose elements may get either true or false values. This matrix can be formed by the following a step-by-step procedure:

1. Starting from position (1,2) in the first row of the matrix, put in this row the names of the primary descriptors of set T
2. Starting from position (2,1) in the first column of the matrix, put in this column the names of the dependent descriptors of set S
3. For the row corresponding to s_k, set the value true (T) in the column of t_u if the name of t_u exists in T_k, otherwise set false (F)
4. Repeat step 3 for all the columns
5. Repeat steps 3-4 for all the rows

Now, the matrix should look as follows (in the example matrix below the Ts and Fs have been arbitrarily positioned):

$$\begin{bmatrix} & t_1 & t_2 & t_3 & . & . & . & t_p \\ s_1 & T & F & T & . & . & . & F \\ s_2 & F & T & T & . & . & . & T \\ s_3 & F & T & F & . & . & . & F \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ s_m & T & T & T & & & & F \end{bmatrix} \tag{7}$$

The matrix represents the relations among the design descriptors. It records the existence or non-existence of relations among the members of S and T as they result from an exhaustive formal depth first search in set D.

For every $s_i, i = 1,2,...,m$, the *ratio* between the corresponding elements in I_n and E_n can be calculated:

$$d_i = I(s_i)_n / E(s_i)_n, i = 1,2,...,m \tag{8}$$

The rows of the dependency matrix can be rearranged in descending order according to the values of d_i. Then the uppermost rows will correspond to dependent descriptors that present high importance indices and low design effort. A second arrangement of the matrix rows may be performed if there are groups of rows that present equal d_i values. In that case and for every row group, the rows are rearranged in ascending order of design effort values as they appear in E_n.

For every $t_i, i = 1,2,..., p$, the *ratio* between the corresponding elements in O_n and E_n can be calculated:

$$u_i = O(s_i)_n / E(s_i)_n, i = 1,2,..., p \tag{9}$$

The columns of the dependency matrix can be rearranged in descending order according to the values of u_i. Then the leftmost columns will correspond to primary descriptors that present high occurrence indices and low design effort. A second arrangement of the matrix columns may be performed if there are groups of columns that present equal u_i values. In that case and for every column group, the columns are rearranged in ascending order of design effort values as they appear in E_n. Figure 2 is a qualitative representation of the dependency among the row number (x-axis) and the normalized importance and design effort (y-axis).

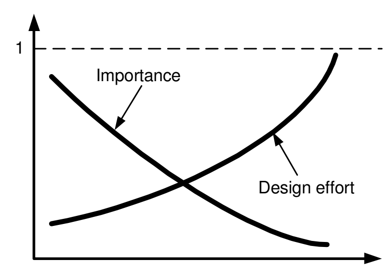


Figure 2. Importance of dependent descriptors and instantiation effort.

2.5 The instantiation list

In [2], an ordered instantiation sequence (list) of primary descriptors was suggested which when followed could ensure the production of design information of maximum importance with the least effort in the early design stages. In the present paper, the top row of the dependency matrix contains primary descriptors of the design problem and a list is also formed. However, in the present case, the formation of this list is strongly affected by the six (6) factors (see page 4) through the relations (1), (2), (8) and (9). The final order of the descriptors is again determined by the repetitive rearrangements of the rows and columns and corresponds to a more realistic treatment of the design knowledge (compared with that followed in [2]) since the instantiation effort is calculated differently for every primary descriptor.

If the designer starts instantiating the primary descriptors according to the order suggested by this list, the most important dependent descriptors are instantiated first. Additionally, the primary descriptors that present the highest possible occurrence are instantiated first and design information of major importance is produced with the least design effort.

The instantiation list for primary descriptors suggests an instantiation order that takes into account the aforementioned factors and corresponds to more design effort per design cycle when compared with the list suggested in [2]. This seems to be reasonable since the latter was formed without any differentiation regarding the instantiation effort for each primary descriptor, representing an "ideal" design situation and, additionally, it was a product of pure mathematical treatment of the design knowledge.

The conformation of the instantiation process with the order dictated by the modified list in the present work will lead to sub-optimum, yet realistic results when compared with the list in [2]. However, the modified list will comprise the optimum instantiation sequence for the given factors and all other instantiation sequences will produce worse results.

CASE STUDY: DESIGN OF A SUNSHIELD FOR A CAR WINDSCREEN

Figure 3 shows a simplified drawing of a motor-driven mechanism that moves the windscreen sunshield of a passenger car. The body of the sunshield may be rotated - within a range of allowed angle values - around a horizontal axis in order to provide frontal sun protection. The rod – in its turn – may be also rotated around a vertical axis in order to provide side protection to the car passenger. When the sunshield is in its normal position, its rod remains clamped at its right end. An electric motor provides power for these movements that are realized via a pair of splined parts and two pairs of bevel gears (see Figure).

In order to represent the major design entities, one hundred and six (106) descriptors in total are used (see Table 1). Twelve (12) descriptors are primary (they appear underlined in the table).

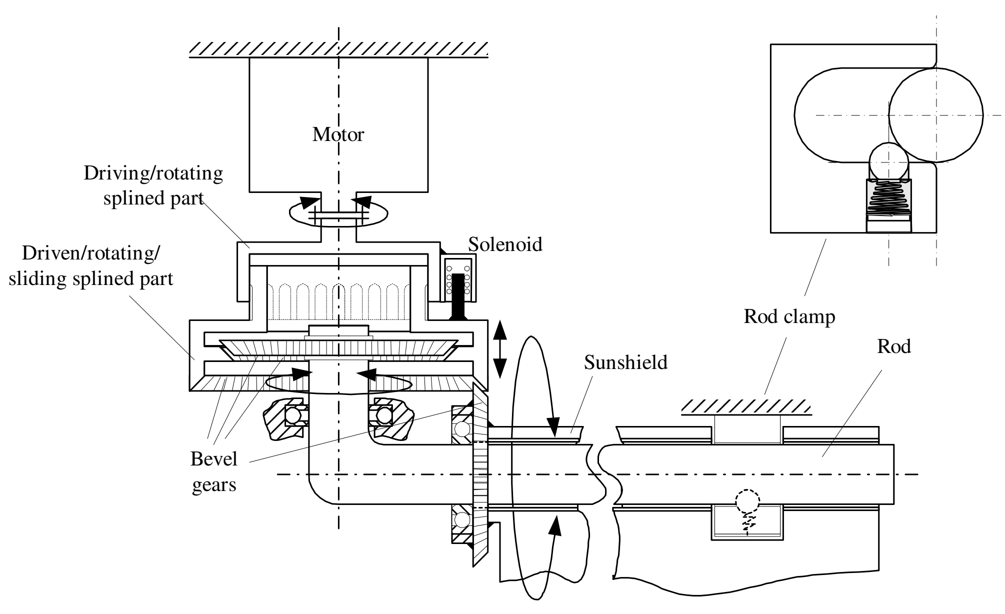


Fig.3. A mechanism providing motion for the windscreen sunshield of a passenger car.

Table 1: Descriptors for the design of a sunshield for a car windscreen

1. ***Sunshield body*** (SB)
- Length (SB.L)
- Left-Part Length (SB.LPL)
- Right-Part Length (SB.RPL)
- Height (SB.H)
- Thickness (SB.T)
- Rotation Angle (SB.RA)

2. ***Sunshield Through Cylinder*** (STC)
- Length (STC.L)
- Internal Diameter (STC.ID)
- External Diameter (STC.ED)

3. ***Sunshield Bevel Gear*** (SBG)
- Position relative to Rod (SBG.PRR)
- Face Width (SBG.FW)
- Pitch Diameter (SBG.PD)
- Diametral Pitch (SBG.DP)
- Number of Teeth (SBG.NT)
- Gear Axis (SBG.GA)
- Torque (SBG.T)
- Gear Ratio (SBG.GR)

4. ***Sunshield Bearing Nest*** (SBN)
- Position relative to Rod (SBN.PRR)
- Internal Diameter (SBN.ID)
- External Diameter (SBN.ED)
- Height (SBN.H)

5. ***Sunshield Ball Bearing*** (SBB)
- Width (SBB.W)
- Position relative to Rod (SBB.PRR)
- Internal Diameter (SBB.ID)
- External Diameter (SBB.ED)

6. ***Rod*** (R)
- Horizontal Part Length (R.HPL)
- Vertical Part Length (R.VPL)
- Diameter (R.D)
- Rotation Angle (R.RA)
- Rotation Speed around Horizontal Axis (R.RSHA)
- Rotation Speed around Vertical Axis (R.RSVA)

7. ***Rod Thrust Bearing*** (RTB)
- Width (RTB.W)
- Position relative to Rod (RTB.PRR)
- Internal Diameter (RTB.ID)
- External Diameter (RTB.ED)

8. ***Rod Bevel Gear*** (RBG)
- Position relative to Rod (RBG.PRR)
- Face Width (RBG.FW)
- Pitch Diameter (RBG.PD)
- Diametral Pitch (RBG.DP)
- Number of Teeth (RBG.NT)
- Gear Axis (RBG.GA)
- Torque (RBG.T)
- Gear Ratio (RBG.GR)

9. ***Rod Clamp Body*** (RCB)
- Position relative to Rod (RCB.PRR)
- Height (RCB.H)
- Face Length (RCB.FL)
- Width (RCB.W)
- Notch Height (RCB.NH)
- Notch Face Length (RCB.NFL)
- Notch Depth (RCB.ND)
- Notch Internal Diameter (RCB.ID)
- Upper Diameter of Ball Hole (RCB.UDBH)
- Lower Diameter of Ball Hole (RCB.LDBH)
- Metric Thread Size for Retaining Bolt (RCB.MTSRB)

10. ***Rod Clamp Retaining Ball*** (RCRB)
- Diameter (RCRB.D)
- Retaining Plate Length (RCRB.RPL)

11. ***Rod Clamp Spring*** (RCS)
- Spring Material (RCS.M)
- Diameter (RCS.D)
- Length (stretched) (RCS.SL)
- Length (non-stretched) (RCS.NSL)
- Spring Constant (RCS.SC)
- Required Retaining Force (RCS.RRF)

12. ***Rod Clamp Spring Retaining Bolt*** (RCSRB)
- Metric Thread Size (RCSRB.MTSRB)
- Length (RCSRB.L)

13. **Rod Clamp Spring Lock Washer** (RCSLW)
- Diameter (RCSLW.D)
- Height (RCSLW.H)

14. ***Driven Part with one spline and two bevel gears*** (DPSTBG)
- Total Length (DPSTBG.TL)
- Upper external diameter (DPSTBG.UED)
- Upper internal diameter (DPSTBG.UID)
- Upper Length (DPSTBG.UL)
- Spline Engagement Length (DPSTBG.SEL)
- Spline Diameter (DPSTBG.SD)
- Spline Number of Teeth (DPSTBG.SNT)
- Spline Tooth Height (DPSTBG.STH)
- Spline Internal Diameter (DPSTBG.SID)
- Spline Tooth Gap (DPSTBG.STG)
- Lower external diameter (DPSTBG.LED)
- Lower internal diameter (DPSTBG.LID)
- Lower Length (DPSTBG.LL)
- Upper Bevel Gear Position relative to Driven Part (DPSTBG.UBGPRDP)
- Upper Bevel Gear Face Width (DPSTBG.UBGFW)
- Upper Bevel Gear Pitch Diameter (DPSTBG.UBGPD)
- Upper Bevel Gear Diametral Pitch (DPSTBG.UBGDP)
- Upper Bevel Gear Number of Teeth (DPSTBG.UBGNT)
- Upper Bevel Gear Torque (DPSTBG.UBGT)
- Upper Bevel Gear Gear Ratio (DPSTBG.UBGGR)
- Lower Bevel Gear Position relative to Driven Part (DPSTBG.LBGPRDP)
- Lower Bevel Gear Face Width (DPSTBG.LBGFW)
- Lower Bevel Gear Pitch Diameter (DPSTBG.LBGPD)
- Lower Bevel Gear Diametral Pitch (DPSTBG.LBGDP)
- Lower Bevel Gear Number of Teeth (DPSTBG.LBGNT)
- Lower Bevel Gear Torque (DPSTBG.LBGT)
- Lower Bevel Gear Gear Ratio (DPSTBG.LBGGR)

15. ***Driving Part with one spline*** (DPS)
- Length (DPS.L)
- External diameter (DPS.ED)
- Spline Engagement Length (DPS.SEL)
- Spline Diameter (DPS.SD)
- Spline Number of Teeth (DPS.SNT)
- Spline Tooth Height (DPS.STH)
- Spline Internal Diameter (DPS.SID)
- Spline Tooth Gap (DPS.STG)

16. ***Windscreen*** (W)
- Available length (W.AL)
- Vision angle (W.VA)
- Mirror Position (W.MP)

17. ***Motor*** (M)
- Rotation Speed (M.RS)
- Shaft Diameter (M.SD)

Table 2: Factor values for the primary descriptors

	r_{da}	r_{pc}	r_{cs}	r_{st}	r_{cw}	r_{ra}	*Sum*
R.RSVA	8	8	10	10	10	5	51
RCS.RRF	10	5	8	3	10	2	38
R.RSHA	8	10	10	10	10	8	56
SBG.T	10	10	10	10	10	10	60
W.AL	2	1	2	2	1	1	9
W.MP	1	1	1	1	1	1	6
R.D	6	6	5	3	6	4	30
W.VA	10	10	7	6	10	5	48
M.SD	6	4	1	1	1	3	16
RCS.M	1	3	1	1	1	3	10
RCSLW.H	7	4	1	1	4	4	21
RCSRB.L	3	2	1	1	1	3	11

Case 1: First it is assumed that the primary descriptors present equal instantiation effort. This implies that the arrangement of the elements of the instantiation list is guided only by the occurrence values of the primary descriptors (see relation (9)). By applying the method described in sections 2.4 and 2.5, the following list is produced:

{R.RSVA, RCS.RRF, R.RSHA, SBG.T, W.AL, W.MP, R.D, W.VA, M.SD, RCS.M, RCSLW.H, RCSRB.L}

If the designer follows this list, then he/she must instantiate the primary descriptors with the following order:

1. Rotation Speed around Vertical Axis, 2. Required Retaining Force in the Rod Clamp, 3. Rotation Speed around Horizontal Axis, 4. Applied Torque on the Sunshield Bevel Gear, 5. Windscreen Length, 6. Position of the Mirror on the Windscreen, 7. Rod Diameter, 8. Vision Angle for the Driver, 9. Diameter of the Motor Shaft, 10. Material for the Spring in the Rod Clamp, 11. Height of the Lock Washer in the Rod Clamp, 12. Length of the Retaining Bolt in the Rod Clamp

Case 2: Next, it is assumed that each one of the twelve primary descriptors presents different values for every affecting factor (in a 1-10 scale). These values are shown in Table 2 and represent: a. the current knowledge that the designer and/or design team has for each primary descriptor regarding its meaning as well as its instantiation process and b. objective facts about the complexity of the procedure needed for this instantiation, about the constraints and the standards that restrict the descriptor values, etc. By taking into account the values of Table 2 and by applying the method described above (see sections 2.4 and 2.5), the instantiation list is modified as follows:

{W.MP, W.AL, RCS.RRF, R.RSVA, R.RSHA, SBG.T, R.D, M.SD, RCS.M, W.VA, RCSLW.H, RCSRB.L}

Now, if the designer follows this list, he/she must instantiate the primary descriptors with the following order:

1. Position of the Mirror on the Windscreen, 2. Windscreen Length, 3. Required Retaining Force in the Rod Clamp, 4. Rotation Speed around Vertical Axis, 5. Rotation Speed around Horizontal Axis, 6. Applied Torque on the Sunshield Bevel Gear, 7. Rod Diameter, 8. Diameter of the Motor Shaft, 9. Material for the Spring in the Rod Clamp, 10. Vision Angle for the Driver, 11. Height of the Lock Washer in the Rod Clamp, 12. Length of the Retaining Bolt in the Rod Clamp

In this new list, it is obvious that the positions of certain descriptors have changed due to the modification of the design effort indices caused by the factor values.
Figure 4 contains four figures that present some design metrics for the design problem under consideration and demonstrate the difference between the two cases. Figure 4.a shows the normalized

difference of design effort for the dependent descriptors of the final dependency matrix. It is obvious that for the case of differentiated amount of instantiation effort for every primary descriptor, the required design effort remains high throughout the design process (short dash dot line). On the contrary, the importance of each individual dependent descriptor (see Figure 4.b) does not seem to be affected. In this figure and in order to preserve its clarity the importance is plotted for the first 50 dependent descriptors only. For the rest of them the importance tends to zero in both cases.
Regarding the total information importance some deviations are observed. For the case of differentiated amount of instantiation effort for every primary descriptor, there is a continuous delay in the generation of information importance up to the sixth cycle (see Figure 4.c) caused by the reordering of the primary descriptor list (see above) that has "pushed back" – due to large amount of design effort needed for their instantiation- primary descriptors that could, otherwise, instantiate dependent descriptors of greater importance. The same behaviour is observed for the percentage of the instantiated dependents per design cycle (see figure 4.d). There is a delay in the production of dependent descriptors for all design cycles.

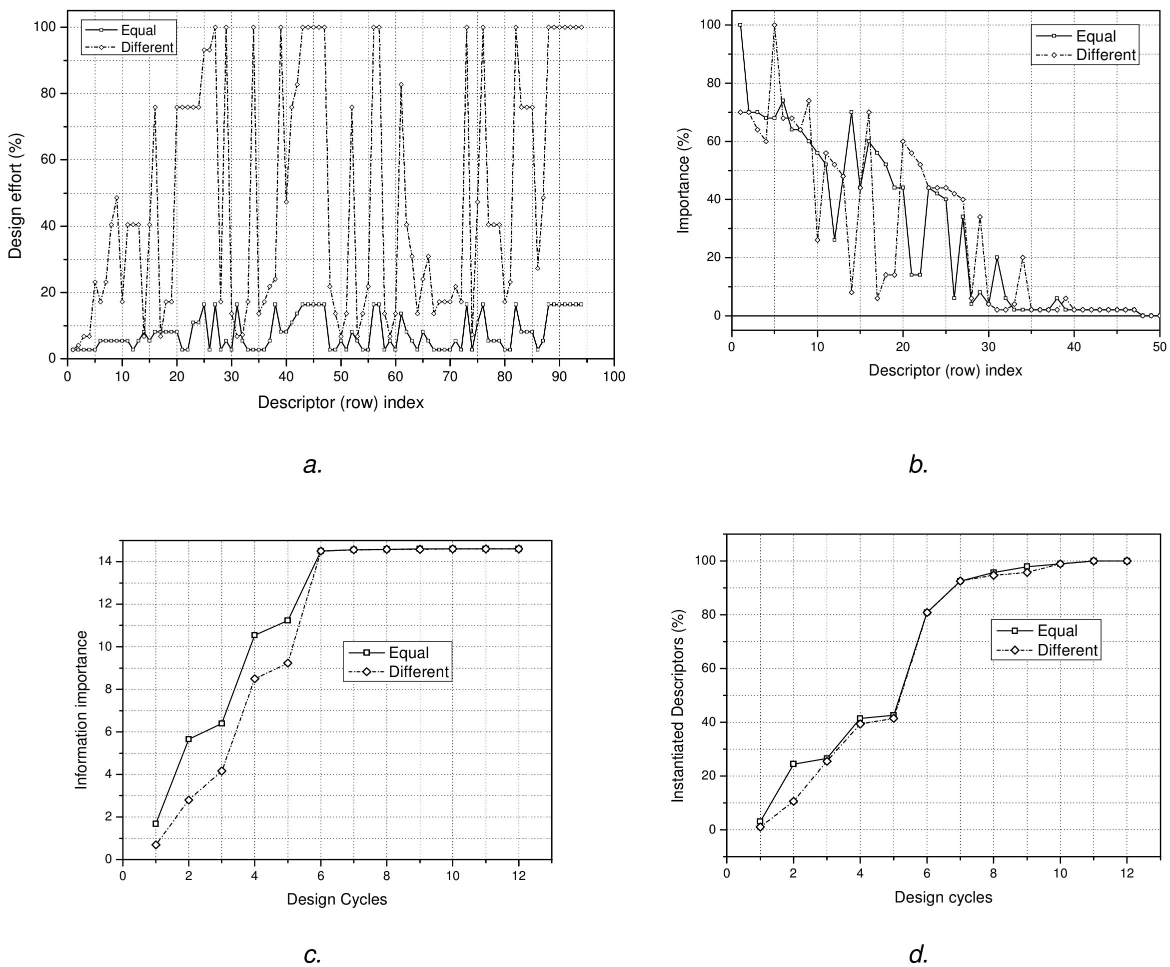


Figure 4. Design metrics for the case of a windscreen sunshield.

CONCLUSIONS

The need for production of high importance information during the very first design phases is obvious for all those that consider design as a process characterized by repetitive decision-making steps. The designer can always make better decisions concerning the design route to be followed and the design solutions to be adopted if adequate design information of high importance is promptly available.

The current work is a contribution towards this direction. It handles additional information for every primary descriptor participating in a design via factors that represent subjective and objective data and provide qualitative measures about the required instantiation effort for its instantiation. The combination of the mathematical, problem – independent approach of design studied in previous papers with the handling of this additional information presented in the present paper, provides a more realistic approach of the design process.

The proposed methodology may form a basis for building further designer-assisting tools that could be used in case of large design problems where a great number of descriptors are used. In this case, the descriptor instantiation process could be optimized so that the most important information is produced early within the design process and with least design effort. Although already validated through a series of small design cases, the method will be tested in large-scale design problems in the very near future.

ACKNOWLEDGMENT

University of Patras is a member of the EU-funded I*PROMS Network of Excellence.

REFERENCES

[1] Drakatou S., Dentsoras A.J., "A Method for the Automatic Deduction of Priority Lists of Entities and Tasks from the Design Knowledge," Artificial Intelligence in Engineering, Design, Analysis and Manufacturing, 15(3), (2001), p. 223-232

[2] Dentsoras, A.J., "Information Generation during Design: Information Importance and Design Effort," Artificial Intelligence in Engineering, Design, Analysis and Manufacturing,19, (2005), p. 19-32

[3] Shooter, S.B., Keirouz, W.T., Szykman, S., Fenves, "A Model for the Flow of Design Information in Product Development," Engineering with Computers, 16, (2000), p. 178-194,

[4] Lee, C-H., Sause, R., Hong N.K., 'Overview of entity-based integrated design product and process models," Advances in Engineering Software, 29(10), (1998), p. 809-823

[5] Warfield, J. N.,"Binary Matrices in System Modeling," IEEE Transactions on Systems, Man, and Cybernetics, 3, (1973), p. 441-449

[6] Steward, D. V., "The Design Structure System: A Method for Managing the Design of Complex Systems," IEEE Transactions on Engineering Management, 28, (1981), p. 71-74

[7] Kusiak, A. & Larson, N., "Decomposition and Representation Methods in Mechanical Design," ASME Transactions: Journal of Mechanical Design, 117(3), (1995), p. 17-24

[8] Kusiak, A. & Park, K., "Concurrent Engineering: Decomposition and Scheduling of Design Activities," International Journal of Production Research, 28(10), (1990), p. 1883-1900

[9] Kusiak, A. & Wang, J., "Decomposition of the Design Process,"Journal of Mechanical Design, 115, (1993), p. 687-695

[10] Eppinger, S. D., " Model-based Approaches to Managing Concurrent Engineering," Journal of Engineering Design, 2, (1991), p. 283-290

[11] Black, T. A., Fine, C. H., Sachs, E. M., " A Method for Systems Design Using Precedence Relationships: An Application to Automotive Brake Systems," M.I.T. Sloan School of Management, Cambridge, MA, (1990), Working Paper no. 3208

[12] Austin, S., Baldwin, A., Newton, A., "Manipulating the flow of design information to improve the programming of building design," Construction Management & Economics, 12(5), (1994), p. 445-455

[13] Yassine, A., Falkenburg, D., Chelst, K., "Engineering Design Management: An Information Structure Approach," Int. J. Prod. Res., 37(13), (1999), p. 2957-2975

[14] Browning, T. R., "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," IEEE Transactions On Engineering Management, 48(3), (2001), p. 292-306

[15] Bashir, H. A., Thomson, V., "Metrics for design projects: a review, " Design Studies (20), (1999), p. 263-277

[16] Bashir, H. A., Thomson, V., "Models for estimating design effort and time," Design Studies, (22), (2001), p. 141-155

[17] Bashir, H. A., Thomson, V., "An analogy-based model for estimating design effort," Design Studies (22), (2001), p. 157-167
[18] Hölttä, K.M.M., Otto, K.N., "Incorporating design effort complexity measures in product architectural design and assessment," Design Studies, (26), (2005), p 463-485
[19] Dentsoras, A.J., "An Approach of Routine Design Based on Extensive Design Space Search," Proc. IITT-Inter. Conf. EXPERSYS-96, (1996)
[20] Tsalidis, S. S., Dentsoras, A. J., "Application of Design Parameters Space Search for Belt Conveyor Design," Engineering Application of Artificial Intelligence, 10(6), (1997)., p. 617-629

Contact: A.J. Dentsoras
University of Patras
Department of Mechanical Engineering and Aeronautics
Patras, 26500
Greece
+30 2610 997217
+30 2610 997217
dentsora@mech.upatras.gr
http://www.mech.upatras.gr/~dentsora