Transdisciplinary Lifecycle Analysis of Systems
R. Curran et al. (Eds.)

doi:10.3233/978-1-61499-544-9-595

The Design Process Structural & Logical Representation in the Concurrent Engineering Infocommunication Environment

Denis TSYGANKOV[1], *Alexander POKHILKO, Andrei SIDORICHEV, Sergey RYABOV, Oleg KOZINTSEV*
Ulyanovsk State Technical University, 32 North Venets st., 432027 Ulyanovsk, Russian Federation

Abstract. Design of composite objects differs by the presence of the technical requirements to the entire device, without separate requirements to the components. Results of the design activity are the technical documentation and a 3D-assembly. Generation of a 3D-assembly includes adding 3D-models of components and the introduction of conjugations between them. For 3D-assembly corrections and adjustments, it is needed to replace manually the parts and install their conjugations that take considerable time. The proposed solution is to establish structural and logical relationships between the components of the 3D-assembly model. This allows components to bind with structural elements of each other. Interconnections between the components can determine their location and conjugation with each other, as well as to modify and export the final design solutions, not disrupting its integrity, what is not provided by any of the modern CAD systems.

Keywords. CAx technologies, design activities, automation, design solutions, 3D-image, interoperability, design process.

Introduction

Accomplishment of interoperability remains a major challenge. It complicates the complete exchange of project activities results between different CAD systems. However, currently offered approaches for solving this problem cannot be implemented for a various reasons [1], [2]. The theoretical basis of this research is presented and described.

Design solution representation based on technology of structural logic linking of design procedures and using free geometric kernel Open CASCADE Technology is an approach that allows us to solve the problem of interoperability in systems of 3D-design, and ensure the preservation of the structural and logical integrity of design decisions [3]. While the solutions are described as a set of techniques required for their formation [4].

[1] Corresponding author, E-mail: furius73@gmail.com

1. Design process procedural representation

The design process can be represented by a sequence of design procedures, each of which has specific physical meaning. The simplest option – linear performance of design stages is shown in Fig. 1. (Left).

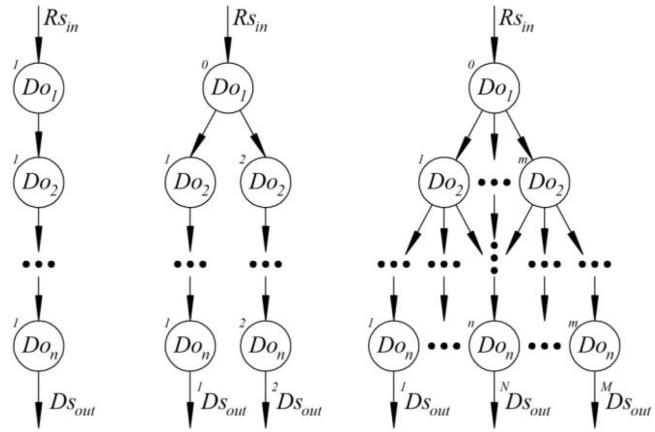

Figure 1. Design process procedural representations.

The Scope Statement comes first stage Rs_{in}, which is a set of input data and the requirements for future product. Each next step is carried out after successful completion of the previous one. Input data for i-th step are output (i-1)-th step. Moreover, steps may have different variants of transforming source data into output data; it is called an branches of alternative.

For a linear sequence of steps, the Ds_{out} design solution formed by the formula:

$$Ds_{out} = (Dpo_1, 1) \cup (Dpo_2, 1) \cup \ldots \cup (Dpo_n, 1); \quad (1)$$

Various solutions can be prepared by the following sequence of design steps represented by the formula (1). However, this sequence is able to describe a class of geometric objects such as variability does not provide for completion of design procedures. The sequence, which has two variants for the design routes of the *1-st* step of the project is shown in Figure 1. (in the center).

Thus, the values of input and intermediate design parameters determine the subsequent design stage – $Do^1{}_2$ or $Do^2{}_2$, each of them will lead to the original design solution $Ds^i{}_{out}$.

For this sequence, the design solution Dsout expressed by the formula:

$$Ds_{out} = \{[(Dpo_1, 1) \cup \ldots \cup (Dpo_n, 1)] \cap [(Dpo_1, 2) \cup \ldots \cup (Dpo_n, 2)]\}; \quad (2)$$

Building a 3D-model of the cylindrical detail can be considered as a typical example: it can be obtained as the different software as well as different ways (by drawing a sketch, its rotation, built a loft, and others.).

The most general case – the design tree sequence of steps, each of which has several branches alternatives presented in Figure 1. (Right).

In the general case, the set of possible treatments n is formed as a result of i-th step of the project, on $(i + 1)$-th stage. As a result, this sequence allows to obtain the set of design solutions at output, each of them is unique, but takes into account the specifics of the project tree routes (like the design of objects in a single class).

In the general design solution, Ds_{out} formed according to the following formula:

$$\begin{aligned} Ds_{out} = (Dpo_{1}, 0) \cup \{ [(Dpo_{2}, 1) \cup ((Dpo_{3}, 1.1) \cap \ldots \cap (Dpo_{n}, 1.n))] \cap \\ \cap [(Dpo_{2}, 2) \cup ((Dpo_{3}, 2.1) \cap \ldots \cap (Dpo_{n}, 2.n))] \cap \ldots \cap \\ \cap \ldots \cap [(Dpo_{2}, m) \cup ((Dpo_{3}, m.1) \cap \ldots \cap (Dpo_{n}, m.n))] \} \end{aligned} \quad (3)$$

Since each branch of the project route can be characterized by certain property, that distinguishes one solution from the other, the represented sequence may be used in the design of complex technical objects.

2. Design procedures structural & logical representation

The initial data for the design is scope statement that can be described as a set of input design parameters:

$$Rs_{in} = \{ (dp_1^N, dp_2^N \ldots dp_n^N), (dp_1^Q, dp_2^Q \ldots dp_m^Q) \}, \quad (4)$$

Rs_{in} – set of input design parameters dp^N_i и dp^Q_i – design parameters, taking the quantitative and qualitative values. Scope statement defined by the user.

Scope statement defined by the user. The parameters dp^N_i and dp^Q_i can be set by interactive input of arbitrary values as well as the choice of normalized values. This occurs because to the fact that they are limited by various conditions. Thus, dp^N_i parameters can take a range of values $[dp^N_{i.min} \ldots dp^N_{i.max}]$, whereas dp^Q_i parameters – only discrete values $[dp^Q_1, dp^Q_2 \ldots dp^Q_n]$. Taken values of parameters both types are predefined according to the design algorithm and other standards.

The resulting 3D-model is generated according output design parameters:

$$Rd_{out} = \{ (dp_1^A, dp_2^A \ldots dp_n^A), (dp_1^C, dp_2^C \ldots dp_m^C) \}, \quad (5)$$

Rd_{out} – set of output design parameters, dp^A_i and dp^C_i – design parameters whose values are determined automatically and interactively by the user. Each of the parameters affects the formed solution, making it a unique design for all output parameters.

Automatic determination of the values of output parameters dp^A_i occurs according to a predetermined algorithm – design techniques which contains a minimal required set of initial data on its output. It should be noted that the specific value of the *i-th* parameter affects the decision, as well as the system their values, this factor is used in the selection the optimal structure of design solution. Some output solutions dp^C_i may have several alternative values that correspond to the technical task. Therefore, the user must select the preferred option “manually”.

In general, the output Rd_{out} design parameters are determined by the function:

$$Rd_{out} = f\left(Rs_{in}, Rs_{md}, Com_n^f\right), \quad (6)$$

Rs_{md} – a set of intermediate of design parameters, Com^f_n – set of design parameters of *n* components included in the designed device. The set of parameters Com^f_n completely determines the three-dimensional images of all the components of the composite device.

When designing any device is represented as a system of interconnected components, therefore it can be formally submitted in the following form:

$$RtD_{Form} = \{ Com_1^f, Int_1^f, Com_2^f, Int_2^f \ldots Com_n^f, Int_n^f \}, \quad (7)$$

RtD_{Form} – formal representation of a composite device, Int^f_i – is a set of interconnections of design parameters of *i-th* component.

The set design parameters of (Com^f_i) components similar to the formula (1):

$$Com_1^f = \{ (dpc_1^N, dpc_2^N \ldots dpc_n^N), (dpc_1^Q, dpc_2^Q \ldots dpc_m^Q) \}, \quad (8)$$

$dpc^N i$ and $dpc^Q i$ – design parameters, taking the numerical and qualitative values.

The set $Int^f i$ includes three sets: interconnections with input and output design parameters $f(Rs_{in})$ and $f(Rd_{out})$, and interconnections with parameters of other components $f(Com^f i)$:

$$Int_i^f = \{ f(Rs_{in}), f(Rd_{out}), f(Com_1^f \ldots Com_n^f) \}, \qquad (9)$$

Interconnections with the input parameters $f(Rs_{in})$ determine each of the components and arrange them into system in accordance with terms of reference. Interconnections with output parameters $f(Rd_{out})$ provide qualitative and quantitative changes in the system when the terms of reference changed. Interconnections with the parameters of other components $f(Com^f_i)$ allow match the components, ensuring the integrity of design solutions.

In general, the design process is a set of design procedures that display it as a sequence of steps with physical meaning. When designing, there are two design procedures: first one – design procedure for constructing 3D-image of the *i-th* component dP^{3D}_i, and the second one – the design procedure for establishing interrelations between the *i-th* and the *n-th* components of $dP^{Int}_{i,n}$. In this case, the design process Des^P_{Form} can be formally represented in the following form:

$$Des^P_{Form} = \{ (dn_1, da_1, dP_1^{3D}) \ldots (dn_n, da_n, dP_n^{3D}), (dP^{Int}_{1,2} \ldots dP^{Int}_{(n-1),n}) \}, \qquad (10)$$

dn_i – the serial number of execution of the *i-th* project procedure $(dP^{3D}i)$, da_i – branch number of alternative, which owns the current design process. Number of branches alter the native project determines the route in accordance with the specified design parameters, making a unique solution.

3. Approbation

Waveguide horn antenna is considered for example. It includes components such as the open end of the waveguide sectorial E- and H-horns, pyramidal and wedge-shaped horns (Fig. 2).

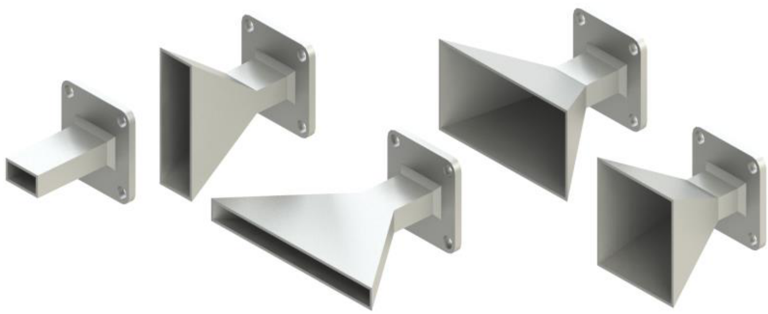

Figure 2. Horn waveguide antennas.

Initially, design parameters common for all components of this devices class must be selected. In the case of waveguide horn antennas, there are the height and width of the waveguide section $a \times b$, flange type T_f, the waveguide section length L_w, the wall thickness of the waveguide t_w, the material m and the type of deposition s_m. Then, the parameters peculiar for some objects are selected. In this case of sectorial and tapered horn there are width of the horn in the E- and H-plane r_E, r_H, and the length of the horn in a plane l_E, l_H. There are same parameters for pyramidal horn, but in this case instead the length of the horn in a particular plane has a length in both planes l.

When all parameters are set, their values are checked. The fact that some parameters may take the whole interval of values, and the other – on only discrete values. Since all input parameters specified by the user, they are limited to certain conditions. In case set values fall in the range, arbitrary values entered by the user interactively $[P_{min} \dots P_{max}]$ (the difference between the two closest values depend on the entered sampling step Δt); and discrete values selected by the user interactively from a predefined "normalized" values, strict adherence between selected option and the offered one is required condition in this case.

Normalized values are determined by reference and scientific and technical literature, which is applicable to the planned technical objects. Horn antennas are part of the microwave waveguide path, and hence the value of the normalized design parameters are determined in accordance with Russian national standards – GOST 20900-2015 and GOST 13317-89.

Table 1 shows the design parameters of the waveguide horn antennas, their lettering and type of input values.

Table 1. Design parameters of horn waveguide antennas class.

Design parameters	Symbol	Input type
The waveguide height and width	$a \times b$	Selection of normalized values of design parameters
Flange type	Tf	
Material	m	
Type of deposition	sm	
The waveguide wall thickness	tw	
The waveguide length	Lw	Interactive input of design parameters values
The E-plane horn width	rE	
The H-plane horn width	rH	
The E-plane horn length	lE	
The H-plane horn length	lH	
The both planes horn length	l	

After the input type of input parameter values specified, the formal representation of the design process device class is formed. It's a series of design procedures that form the design process. Formal representation of the class horn waveguide antenna has the form:

$$\begin{gathered} FrMd_{Horn}^{Ant.} = \{(di_1, dv_1, Dpc_1^{a\times b}), (di_2, dv_2, Dpc_2^{T_f}), (di_3, dv_3, Dpc_3^{m}), (di_4, dv_4, Dpc_4^{s_m}), \\ (di_5, dv_5, Dpc_5^{t_w}), (di_6, dv_6, Dpe_6^{L_w}), (di_7, dv_7, Dpe_7^{r_E}), (di_8, dv_8, Dpe_8^{r_H}), (di_9, dv_9, Dpe_9^{l_E}), \\ (di_{10}, dv_{10}, Dpe_{10}^{l_H}), (di_{11}, dv_{11}, Dpe_{11}^{l}), (di_{12}, dv_{12}, Dpb_{12}^{Oew}), (di_{13}, dv_{13}, Dpb_{13}^{Hes}), \\ (di_{14}, dv_{14}, Dpb_{14}^{Hhs}), (di_{15}, dv_{15}, Dpb_{15}^{Hp}), (di_{16}, dv_{16}, Dpb_{16}^{Hw}), \end{gathered} \tag{11}$$

$FrMd^{Ant}{}_{.Horn}$ – set of design procedures required to design components horn waveguide antenna; di_i – serial number of the *i-th* execution of project procedures $i = 1 \dots 16$; dv_i – branch number of alternative that contains the *i-th* design procedures; Dpc_i – design procedure for selecting normalized values of the input parameters $a \times b$, T_f, m, s_m, t_w, for $i = 1 \dots 5$; Dpe_i – design procedure for interactive entering values of the input parameters L_w, r_E, r_H, l_E, l_H, l, for $i = 6 \dots 11$; Dpb_i – design procedure for creating 3D components of waveguide antenna: the open end of the waveguide (for $i = 12$), sectorial E- and H-horns (for $i = 13,14$), a wedge-shaped and pyramidal horns (for $i = 15,16$).

Branch number of alternative dv_i is unique for each component of class; it defines the design route. If the project procedures are common, $dv_i = 0$.

Design procedure for selecting normalized values of the input project parameters Dps_i is a function that contains only discrete values of the design parameters on the output. For example in case of interactive entering wavelength value λ, the procedure selects the corresponding height and width of the waveguide section $a\times b$, which is shown in Fig. 3. The number of possible discrete values $a\times b$, corresponding to design routes at the procedure output is limited.

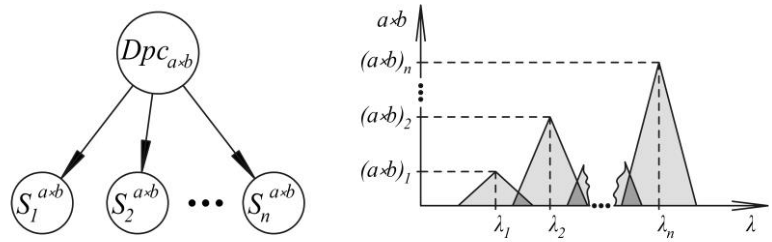

Figure 3. The structure of design procedure for selecting normalized values of the input project parameters.

The design procedure for interactive entering the parameter values Dpe_i – a function that contains a set of parameter values at the output, and their number is controlled by the sampling step Δt – a quantitative measure that distinguishes the two nearest values. For example, when setting the length of the waveguide t_w, the user can enter values from 0 (equivalent to the absence of the waveguide) to $t_{w.max}$. In this case, all the values obtained at the output of the procedures relate to a single project route, but make a unique final design decision. Fig. 4 shows the design decisions associated with one project route, and differ only in the value of design parameters – the width of the aperture of the horn in the H-plane r_H.

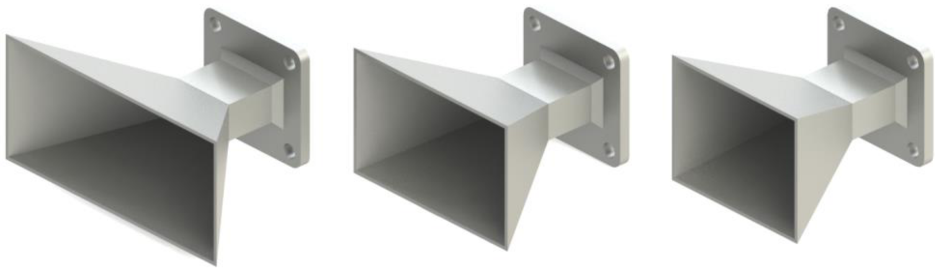

Figure 4. Design solutions corresponding to one branch of the project route.

The design procedure for constructing 3D-image Dpb_i is the main procedure for program implementation. It reflects the principle of the software tools in general. *IDEF0*-model of design procedure for constructing 3D image is shown in Fig. 5.

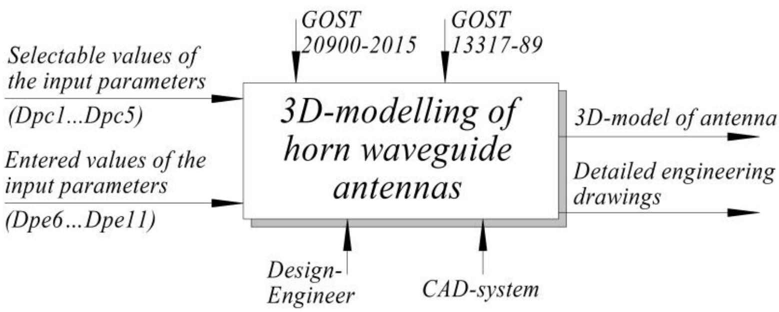

Figure 5. Design procedure for creation 3D-model.

This is a complex procedure; it is an ordered sequence of design procedures. These procedures are functions of the input values (or intermediate values) of design parameters. In addition, each procedure has a serial number and the number of execution of branch alternatives.

Specifying of design procedures included in the procedure for constructing 3D-images, occurs on the basis of structural analysis of all the components that make up the class, which aim – the selection common and unique components that have physical meaning.

Horn antenna includes five components. Common parts: the flange and waveguide segment; unique parts: sectorial horn, pyramidal horn and tapered mouthpiece. Separation of horns explained by the different set of input parameters and design operations required for their construction. Procedural model of designed devices formed after specifying these components.

4. Design process representation based on Open CASCADE Technology

Design procedure for creating 3D-components of horn antennas has the following form:

$$PrcMd^{Ant.}_{Horn} = \{(di_1, dv_1, Dpf_{Fl}), (di_2, dv_2, Dpf_{Wg}), \\ (di_3, dv_3, Dpf_{Sh}), (di_4, dv_4, Dpf_{Ph}), (di_5, dv_5, Dpf_{Wh})\}, \tag{12}$$

$PrcMd^{Ant}_{.Horn}$ – set of design procedures involved in the construction of 3D models of waveguide antenna components; Dpf_{Fl} – design procedure for constructing the flange; Dpf_{Wg} – design procedure for constructing the waveguide; Dpf_{Sh} – design procedure for constructing a sectorial horn; Dpf_{Ph} – design procedure for constructing a pyramidal horn; Dpf_{Wh} – design procedure for constructing a wedge-shaped horn.

Eigen function is formed for each of the component parts. A defined set of design parameters determining its model is the argument of this function. The function of 3D-model of a waveguide flange is presented below as an example.

```
TopoDS_Shape CSAPR_WG_Antenna::Build_Flange (float a, float b)
{
  Set_Size (a, b); // Determination of local parameters
  TopoDS_Shape Base = B_base (a, b, h, r); // Building the base flange
  TopoDS_Shape WaveGuide = B_wg (a, b, Base); // Building the waveguide
  TopoDS_Shape Flange = B_Holes (A, B, a, b, d, h, WaveGuide); // Building holes
  Return Flange;
}
```

Listing 1. Software implementation of design procedures for constructing the flange.

B_Base, B_WG and B_Holes Procedures are code of operations included in the kernel of Open CASCADE libraries. Each function builds a 3D-model completely determined by input parameters, also created models can be combined and subtracted, – all these activities are provided by kernel libraries Open CASCADE.

Thus, there is a distribution of initial design parameters on procedures required for their execution. The design parameters may be initial for all procedures as well as for a single procedure.

Structure of the design solution formation is based on the procedural model. Depending on the selected component, design solution $DSol^{3D}_{Horn}$ is formed by the following formula:

$$DSol^{3D}_{Horn} = \{ (1,0,Dpf_{Fl}) \cup (2,0,Dpf_{Wg}) \cup \\ \cup ((3,1,Dpf_{Sh}) \cap (3,2,Dpf_{Sh}) \cap (3,3,Dpf_{Ph}) \cap (3,4,Dpf_{Wh})) \} \tag{13}$$

As a result of the design project are five different types of solutions can be obtained at the output. They correspond to the number of components of the class shown in Fig. 2. Thus, in some cases, solutions may be identical for different design routes. For example, a wedge horn with the $l_E=l_H$ condition and pyramidal horn with the $l=l_H=L_E$ condition simultaneously.

Despite the fact that the 3D-models are same, solutions relate to various branches design routes. Output values of design parameters (such as beamwidth, gain, etc.) will be the same, confirming the equivalence of these solutions.

5. Conclusions

Program shell is developed based on a formal representation, programmatically: window display the components, in which, after selecting the component output panel set the values of the design parameters on the basis of which formed design solution. Software implementation of formed procedural model is a code of design procedures and operations of the nucleus Open CASCADE Technology, entering into their structure, and their further alignment in the manner determined by the formed design decision.

Structural logic binding of design procedures occurring during software implementation allows designing a class of devices while remaining within the strict framework of rules and algorithms defined by standards and specifications. Consequently it suffices to define the source data – terms of reference, to get the output solution that satisfies it.

The resulting models can be stored in the format of ISO 10303 STEP, strengthen their opening processing and preservation in any modern CAD.

References

[1] L.E. Kamalov and A.F. Pokhilko, The Process Approach to the Synthesis and Analysis of Three-Dimensional Representations of Complex Technical Objects, *Pattern Recognition and Image Analysis*, 2011, Vol. 21, № 3, pp. 491-493.

[2] P. Hamilton, *CAD / CAM / CAE Observer*, 2008, № 2 (38), pp. 34-36

[3] O. Kozintsev, A. Pokhilko, L. Kamalov, I. Gorbachev, D. Tsygankov, A Model for Storing and Presenting Design Procedures in a Distributed Service-oriented Environment, *J. Cha et al. (eds.) Moving Integrated Product Development to Service Clouds in the Global Economy*, IOS Press, Amsterdam, 2014, pp. 84-91.

[4] A.F. Pokhilko, L.E. Kamalov, The Process Approach to Synthesizing and Analyzing of 3D Representations of Complex Technical Objects, *Pattern Recognition and Image Analysis*, 2013, Vol. 23. № 1. pp. 68-73.

[5] M. Sobolewski, *Foreword Next Generation Concurrent Engineering*: Smart and Concurrent Integration of Product Data, Services, and Control Strategies, M. Sobolewski & P. Ghodous (eds) © 2005 ISPE, 620 p.

[6] G. Booch, *Object-oriented analysis and design with applications*, Third Edition. Moscow, Williams, 2008, 720 p.

[7] L.E. Kamalov, A.F. Pokhilko, T.F. Tylaev, A Formal Model of a Complex Estimation Method in Lean Product Development Process. *Proceedings of the 17th ISPE International Conference on Concurrent Engineering*, Springer-Verlag, Lomdon, 2010, pp. 285-289.