Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Design Automation Methodology of a Cube using Solidworks and VB.net

Dwaipayan Roy Chowdhury
(CAD Automation Engineer at Tandemloop Technologies)
B.E, Department of Mechanical Engineering, Dayananda Sagar College of Engineering,
Kumarswamy Layout, Bangalore - 560078

Abstract: **CAD Automation or Design Customization has been at spark after the traditional design practices of '10s. There has been a phenomenal improvement in the design techniques and methodologies about how to design and what to design efficiently. The word "Efficiency" has been a major concern when it talks about various complicated designs. Once the design has been created, designers are required to make changes at a very basic level. This in turn, creates enumerate difficulties at certain sections where designs are made to create from scratch once again. Eventually this idea of time and effort consumption has been referred and continuous improvements have been made in design technology. Hence, design customization comes into the picture. Solidworks designing and modelling software has been used here for writing the macro code for automation. Considering the exponential growth in design methodologies, this paper has depicted works on design automation of a cube on certain parameters. The work also includes description of end to end design methods which makes the author easy to understand the complete process involved in automation. Parameters like dimensions, colors and materials have been considered for automation. After the conceptual development and algorithm build up, an application has been made in order to make the product user friendly.**

Keywords: Design, Automation, Design Customization, Cube design, Color customization, Solidworks Automation, Material automation, Design techniques, VB.Net, Solidworks.

I. INTRODUCTION:

Not many researches have been done on this particular domain as the work mainly involves industrial experts and whatever work is done have confidentiality issues. Hence papers referred is quite few in number. When CAD customization and Design mechanization are coordinates in designing application, a number of points of interest are gotten. With CAD customization, the generation of a drawing and design of a mechanical component can be produced with incredible exactness. Here a test case has been taken i.e. cube, which has been customized on several parameters like dimensions, materials and color of the block. The parameters considered have been selected on the basis of frequently used features of Solidworks which have havoc impact on modelling. Talking about the dimensions, in modern and advanced design techniques, the complete coordinates of a complicated design can be stored in a database and can be accessed using python or other object oriented programming language to process the data i.e. dimensional data and hence change the complete design based on dimensions. Coming to materials, various high-end optimization methods of designing have been evolved where material database can be accessed and materials can be directly called from Solidworks material database. Color being an aesthetic aspect of design which is essentially used when comes about product designing. Hence, all the parameters considered have certain designing aspects and features are often used. Thus automating those would be helpful at great extent [6,7,8].

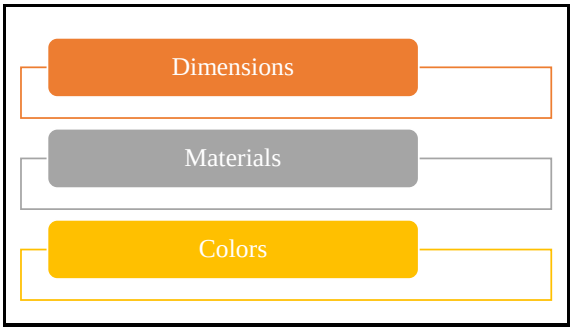


Figure 1.1 (Parameters for Automating)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

II. METHODOLOGY:

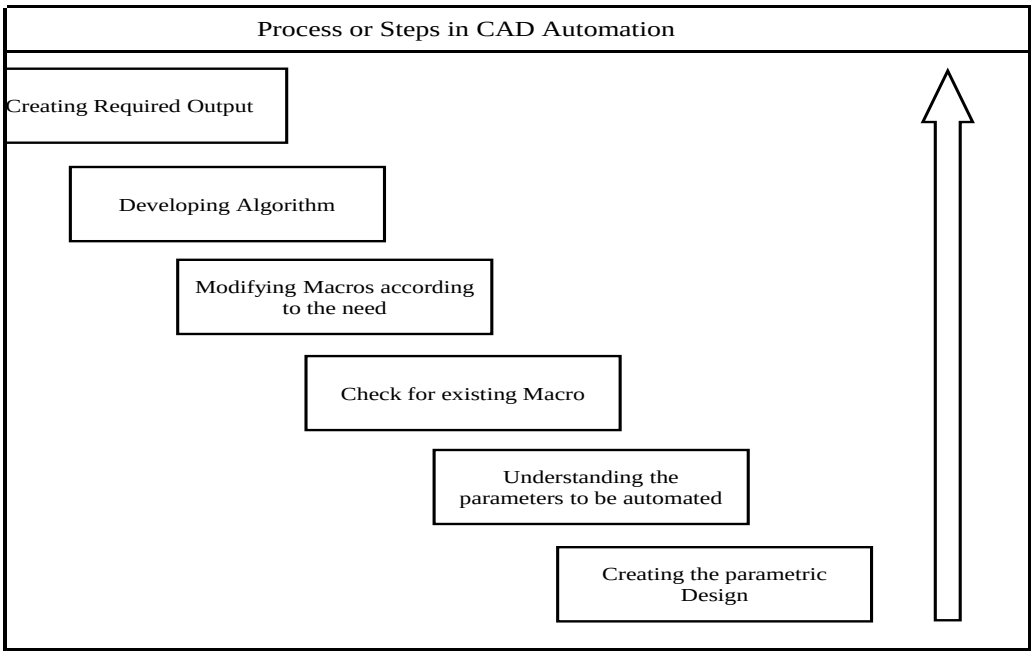


Figure 2.1 (Steps of Automation)

1. Creating the parametric design:

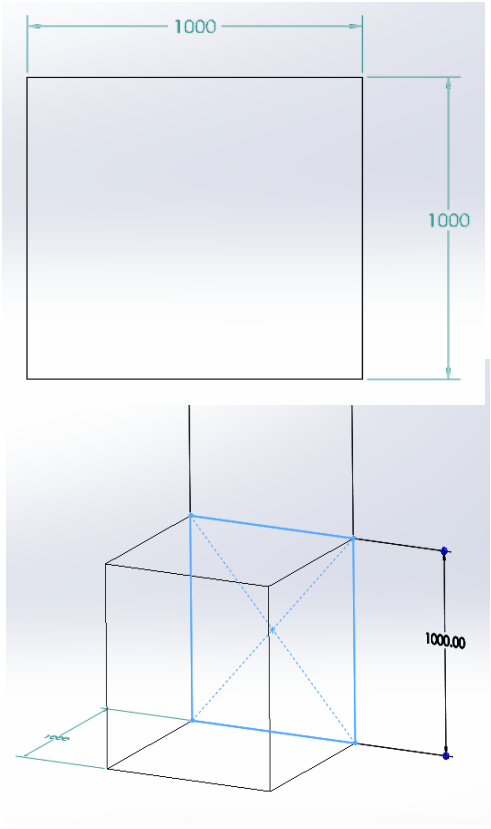


Table 2.1 (Parametric design)

Parametric design involves complete constraining of dimensions after which it develops a relation amongst dimensions. Constraining is very much required in design automation as the customization of dimension for one edge can make others to vary in a slapdash manner [2]. While designing the model, designer has to keep the idea of customizing the parameters and consolidate the id for the following design available in the Solidworks software.

Sometimes, capturing the macro in Solidworks help to get the code for the following parameters, sometimes the algorithm has to be based on self-developed algorithm. [2]

2. Understanding the parameters to be automated:
In the introduction section, it has been described as why the following parameters have been considered. However, understanding the parameters to be automated mainly depends on the client or customer's choice. But often it has been noticed that various other automations are required as by-parts of the major automation like algorithm as to how to rebuild or save the current model or design using small codes [8].

3. Check for existing Macro:
As mentioned earlier, macro capture is one of the most useful feature of Solidworks using which user may get the algorithm for a corresponding feature. User needs to learn how to tactfully use the macro in order to eliminate unnecessary lines for code.

4. Modifying Macros:
After successfully deriving and obtaining the codes of a specific feature from macro, designer needs to focus on using the codes according the specific tasks assigned and needs to consolidate the whole bunch of codes one after the other in a stratified way.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

5. Developing Algorithm:

This stage is the most crucial part of design customization as it holds the complete data management with expertise in programming domain. Developing a suitable algorithm can be monotonous and time consuming at certain stages, but becomes handy when an automation developer becomes a professional with it.

6. Creating required Output:

This phase can be segregated into two subdivisions:

a. Developing the output in Solidworks Macro console.
b. Creating a suitable application incorporating the output macro with certain changes in programming language.

The later one is verily necessary for the user friendly aspect of the user or customer. Clients aren't programmers. Hence they need an application interface where they can make changes in data. [1,5]

III. DIMENSIONAL AUTOMATION:

In the first row, the dimensions have been kept 1 m for length, breadth and height. The dimensions have changed from the existing slapdash dimensions to 1m each sided cube. After that, the dimensions have been moved to length = 2m, width = 3m and height = 4m which converted the 1m each cube to a cuboid with the following dimensions. This change has been monitored in Solidworks software.

3.1 Description:

The algorithm for the dimensions can be directly captured from the Solidworks macro feature. Thus, the algorithm for one edge of the cube is captured at first. It requires the corresponding key for each side's dimensions which is assigned by Solidworks in sequential manner. In the figure 4.1 (Dimension Tab), the marked region is the key for the dimension of one side. Likewise, following dimensions are used in algorithm thereby developing a suitable customized dimensional design.

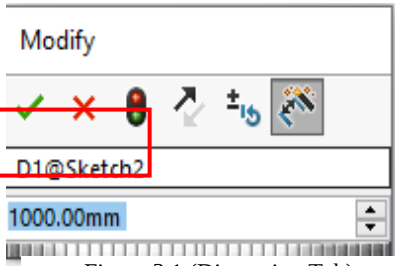


Figure 3.1 (Dimension Tab)

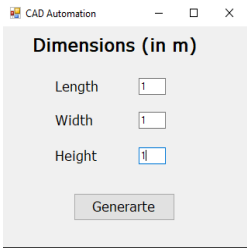
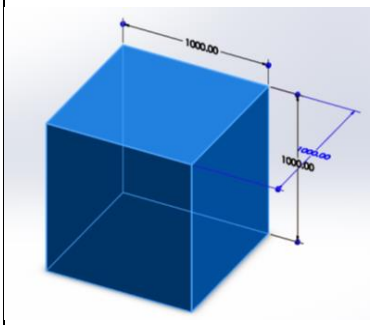
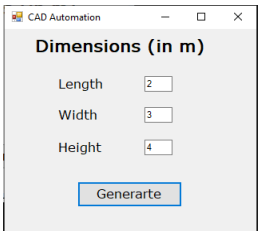
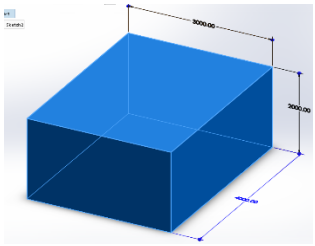
Application	Design in Solidworks
	
	

Table 3.1 Application Interface vs Design Changes (Dimensioning)

3.2 Macro Code:

```
Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Dim myProcess() As Process =
        System.Diagnostics.Process.GetProcessesB
yName ("SLDWORKS")
        swModel                                  =
swApp.OpenDoc6("D:\CAD_Automation\Paper\Cu
be.SLDPRT",

SwConst.swDocumentTypes_e.swDocPART,SwCo
nst.swOpenDocOptions_e.swOpenDocOptions_Sile
nt, "", errors, warnings)
        swModel = swApp.ActiveDoc
        swModelDocExt = swModel.Extension
        swModel.ClearSelection2(True)

        Dim myDimension As Object
        myDimension                              =
swModel.Parameter("D2@Sketch2")
        myDimension.SystemValue                  =
TextBox4.Text
        myDimension                              =
swModel.Parameter("D1@Sketch2")
        myDimension.SystemValue                  =
TextBox5.Text
        myDimension                              =
swModel.Parameter("D1@Boss-Extrude1")
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

```
                myDimension.SystemValue = TextBox6.Text

                swModel.ClearSelection2(True)
                boolstatus1 = swModel.EditRebuild3()
                 swModel.Visible = True

        End Sub
```

The bolded parameter key in the code is the required data from the Solidworks software for dimensional customization. [10]

IV. COLOR CUSTOMIZATION:

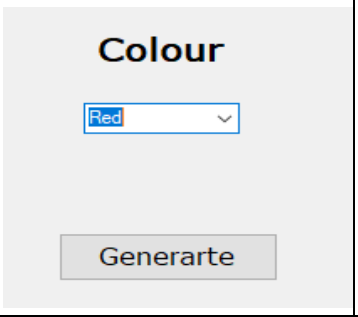
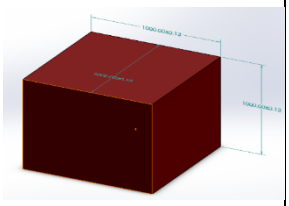
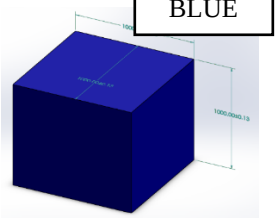
Application Interface	Solidworks design changes
	RED 
	BLUE 

Table 4.1 Application Interface vs Design Changes (Color Customization)

4.1 Description:

The algorithm for the color can't be captured from Solidworks macro. Thus object browser and Solidworks API has been used to develop the code for the same. Its interesting to see that the color change takes place with a property key named self-manager. There material property values are stored in an object variable called vMatprop. And the values corresponding to the first, second and third indices give the color in sequential order as Red, Green, Blue (RGB).

4.2 Macro Code for Application:

```
Private Sub Button1_Click(sender As Object, e Ass EventArgs) Handles Button1.Click
                Dim myProcess() As Process =

        System.Diagnostics.Process.GetProcessesByName("SLDWORKS")
                swModel = swApp.OpenDoc6("#write the directory",
                SwConst.swDocumentTypes_e.swDocPART,
                SwConst.swOpenDocOptions_e.swOpenDocOptions_Silent, "", errors, warnings)
                 If (swModel Is Nothing) Then
                MessageBox.Show("End Previously Running Solidworks File")
                 Else
                swSelMgr = swModel.SelectionManager
                booleanpi = swModel.Extension.SelectByID2("Boss-Extrude1",
                "BODYFEATURE", 0, 0, 0, False, 0, Nothing, 0)
                swComp = swSelMgr.GetSelectedObjectsComponent2(1)
                vMatProp = swComp.MaterialPropertyValues

                 Debug.Print("sdvdsv")

                If (vMatProp Is Nothing) Then '

                        swCompModel = swComp.GetModelDoc
                        If swCompModel Is Nothing Then
                        Debug.Print("Selected component is lightweight; exiting macro.")
                            Exit Sub
                            End If

                        vMatProp = swCompModel.MaterialPropertyValues
                End If

                 If Me.ComboBox1.SelectedIndex = 0 Then
                        vMatProp(0) = 1
                        vMatProp(1) = 0
                        vMatProp(2) = 0

                ElseIf Me.ComboBox1.SelectedIndex = 1 Then
                        vMatProp(0) = 0
                        vMatProp(1) = 1
                        vMatProp(2) = 0
                Else
                        vMatProp(0) = 0
                        vMatProp(1) = 0
                        vMatProp(2) = 1
                End If
                swComp.MaterialPropertyValues = vMatProp

                swModel.ClearSelection2(True)
                End If
                 boolstatus1 = swModel.EditRebuild3()
                swModel.Visible = True
            End Sub [10]
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

V. MATERIAL SELECTION:

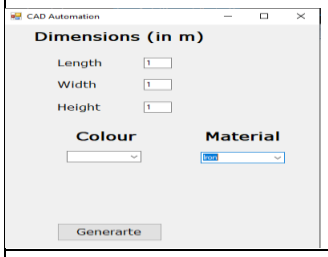
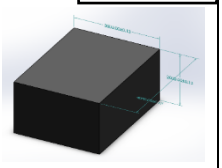
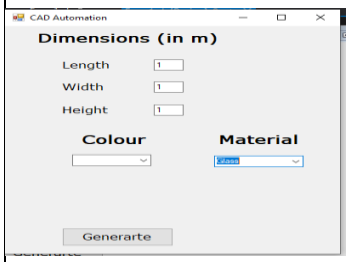
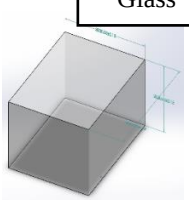
Application Interface	Solidworks Design Changes
	Iron 
	Glass 

Table 5.1 Application Interface vs Design Changes (Material Customization)

5.1 Description:

This is again unique as the material libraries are existent in Solidworks Corps directory. Hence we can directly call the material from the database which is generally located in "C:\Program Files\SOLIDWORKS Corp\SOLIDWORKS\IRay\mdl\solidworks\materials\metal\aluminum\" library. The algorithm for the color can't be captured from Solidworks macro. Thus object browser and Solidworks API has been used to develop the code for the same.

5.2 Macro Code for Application:

```
Dim myProcess() As Process =
System.Diagnostics.Process.GetProcessesB
yName("SLDWORKS")
swApp = New
SldWorks.SldWorks()
swModel = swApp.OpenDoc6("\#mention
the directory ",
SwConst.swDocumentTypes_e.swDocPAR
T,
SwConst.swOpenDocOptions_e.swOpenDo
cOptions_Silent, "", err, warn)
swModelDocExt =
swModel.Extension
swModel.ClearSelection2(True)

If Me.ComboBox2.SelectedIndex = 0 Then
Matvalue1 = "C:\Program
Files\SOLIDWORKS

Corp\SOLIDWORKS\IRay\mdl\solidworks\material
s\metal\aluminum\brushed_aluminum.mdl"
strName = Matvalue1
swAppearance =
swModelDocExt.CreateRenderMaterial(strName)
bool2 =
swAppearance.AddEntity(swModel)
bool2 =
swModelDocExt.AddRenderMaterial(swAppearance,
nDecalID)
Dim swErrors As Long
Dim swWarnings As Long
swModel.Save3(1, swErrors, swWarnings)

ElseIf Me.ComboBox2.SelectedIndex = 1
Then
Matvalue1 = "C:\Program
Files\SOLIDWORKS

Corp\SOLIDWORKS\IRay\mdl\solidworks\material
s\metal\brass\brushed_brass.mdl"
strName = Matvalue1
swAppearance =
swModelDocExt.CreateRenderMaterial(strName)
bool2 =
swAppearance.AddEntity(swModel)
bool2 =
swModelDocExt.AddRenderMaterial(swAppearance,
nDecalID)
Dim swErrors As Long
Dim swWarnings As Long

bool = swModel.Save3(1, swErrors,
swWarnings)
End If
Boolst = swModel.EditRebuild3()

SwModel.ClearSelection2(True)
```

[10]

VI. FUTURE SCOPE:

Here, it has been simplified to 3 major colors which are RGB. But for complex automation, vMatProp has to undergo a loop. Material property also has variety pf options in their database. Extracting the data from their library can also be looped through and can make the automation more dynamic. Hence, making the automation more dynamic can be considered as a future scope for this work.

There are many other features which can be automated like automatic report generation, BOM exportation to excel file, automation in simulation.

VII. CONCLUSION:

Various companies are offering customized products by putting their engineering workforce under tremendous pressure. They are quibbled and squeezed at an extensive proportion to create 2D and 3D documents for the same. This enormous pressure makes the exuberant employees to go wrong at various sections losing jobs and missing deadlines. To avoid these complications and unorganized

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

problems which can't be discern previously, design automation has been inculcated by various industries. In this paper, an elaborate explanation about the complete design automation procedure has been made with examples based on parameters.

Dimensional Automation has been an exclusive customization in this era of automation. For complicated designs which have umpteenth dimensional aspects, database management is required for handling the coordinate values thereby consolidating the complete design in a datasheet. After successful data management, the output data is sent back to the Solidworks software for rebuilding the model.

For color and material automation aspect, there have been phenomenal use of these sort of methodologies in industries. There are n number of materials and color compositions which can be automated by varying the vMatProp within a loop.

VIII. ACKNOWLEDGEMENT:

Design automation reduces company's valuable time by creating automated reports, dimensions, and other features. Thus, this work has been carried out under complete supervision of Sri Harsha L, Director of Tandemloop Technologies and Dr. Ramesha H, associate professor of Mechanical Engineering Department of Dayananda Sagar College of Engineering, Bangalore. They have proctored and overviewed the complete work from scratch till the end. This work is an eclectic collection of knowledge from industrial experts and academician. I would like to extend my sincere regards to both of my mentor, Sri Harsha L and Dr. Ramesha H. I would also like to acknowledge Tandemloop Technologies (which is a CAD Automation company) for sharing minute details about the complete CAD Automation process. It has been a reliable source for me and a complete guide book.

IX. REFERENCES:

[1] Journals:

[1] A Novel Method of Using API to Generate Liaison Relationships from an assembly. Arun Tom Mathew, C.S.P.Rao. 2010. s.l. : Journal of software engg. & Applications, 2010.

[2] Parametric CAD modeling: An analysis of strategies for design Reusabilityl. Camba, Jorge D. 2016. s.l. : Computer-Aided Design 74, 2016, pp. 18-31

[3] Customization of 3D CAD Model For Piston Fixture Using NX Software. Mr. Pratik.S.Koli, Dr. S. K. Patil. 2017. 4, s.l. : International Journal of Scientific & Engineering Research, aPRIL 2017, Vol. 8.

[4] Customization of CAD/CAM software: a case study of customization of UG/NX 4.0 for modeling couplings using knowledge fusion programming. SUNIL D.WAVALE, KISHOR P. KOLHE. 2013. 2, s.l. : ENGINEERING AND TECHNOLOGY IN INDIA, 2013, Vol. 4.

[5] Design and Drawing Automation Using Solid Works Application Programming Interface. Abhishek C. Lad, A.S.Rao. 2014. October 7, 2014, International Journal of Emerging Engineering Research and Technology, p. 158.

[6] Framework for Integrated Mechanical Design Automation, Jouranal of CAD. Hwang, Jack C. H. Chung.Teng-Shang. 2000. 2000.

[7] Kurundkar, R.C. A knowledge driven automation for selection procedure and computer aided design of bearing ; M. Tech. dissertation. s.l. : GGSIE&T.

[8] Marc Saiz Sau. 2010. SolidWorks Parameterization for Industrial Robot Design, Master thesis. s.l. : Department of Management and Engineering , Division of Machine Design, Linkoping's University, Sweden, 2010.

i. [2] Conference:

[9] CAD System Design on Standard Part Based on Software Reuse. J. Tian, S. Liu, and H. Fu. 2011. s.l. : Fourth International Symposium on Knowledge Acquisition and Modeling (KAM), 2011.

[3] Internet Source:

[10] Solidworks API