Master Model Automation Using NX Unigraphics Customization

[1]Jayesh Palekar, [2]S SaravananPillai, [3]Prof. H.P. Khairnar

[1]Student, M.Tech CAD CAM & Automation, [2]Manager-Design, [3]Assistant Professor
[1,3]Mechanical Engineering Department, Veermata Jijabai Technological Institute, Mumbai, India
[2]Heavy Engineering Department, Larsen & Toubro, Powai, Mumbai, India
[1]jcpalekar@gmail.com, [2]s.saravananpillai@larsentoubro.com, [3]hpkhairnar@vjti.org.in

***Abstract* - The parametric model updation in 3D modeling software is very complicated and time consuming process. Updating the assemblies having hundreds of sub-assemblies and parts is very critical job in modern era of CAD-CAM industries. Even a single error in the input can create lots of problem in the final output may be a fatal errors which cannot be retrieve easily. Due to whole Product Lifecycle Management get delayed which cause would cost to company.This paper will illustrate how the customization of the NX Unigraphics models with the help of NX open programing will result reduction of human errors, cycle time, repetitive work and graphics power to enhance design benefits to the company.**

***IndexTerms* - Master Model, Parametric Model, NX Unigraphics, NX Open, Customization, Part Files Updation, API (Application Programing Interface).**

I. Introduction

NX Unigraphics is leading geometric parametric modeling CAD software. It provides a broad set of CAD solutions for the design of complex mechanical products. The Insight NX (Unigraphics) CAD Driver component creates a direct native link to NX, allowing the user to modify CAD geometry, extract relevant output parameters, and export neutral geometry files automatically and without using the CAD system interface. This component can be used together with Insight design drivers to perform NX CAD optimization.

The NX Unigraphics component provides support for native part (PRT) files. It supports input parameters including sketch dimensions, feature dimensions, pattern dimensions, and GD&T dimensions. These entire databases are linked to each other. (Knowledge Fusion Tool of Unigraphics Automation).

Updating these parametric models manually is time consuming and complicated job. For the files having large number of parameters and large space required more time and graphics to process the data. Due to which modeling and designing get delayed, affecting the product lifecycle.

The commonly procedure followed is shown in fig.

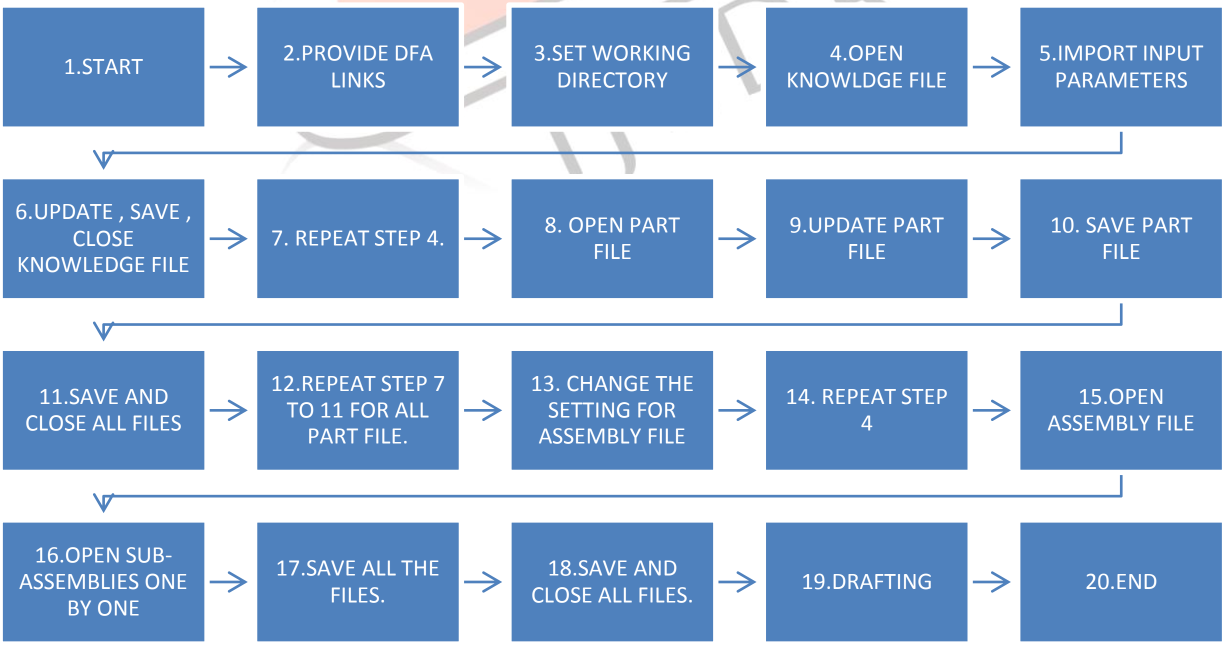
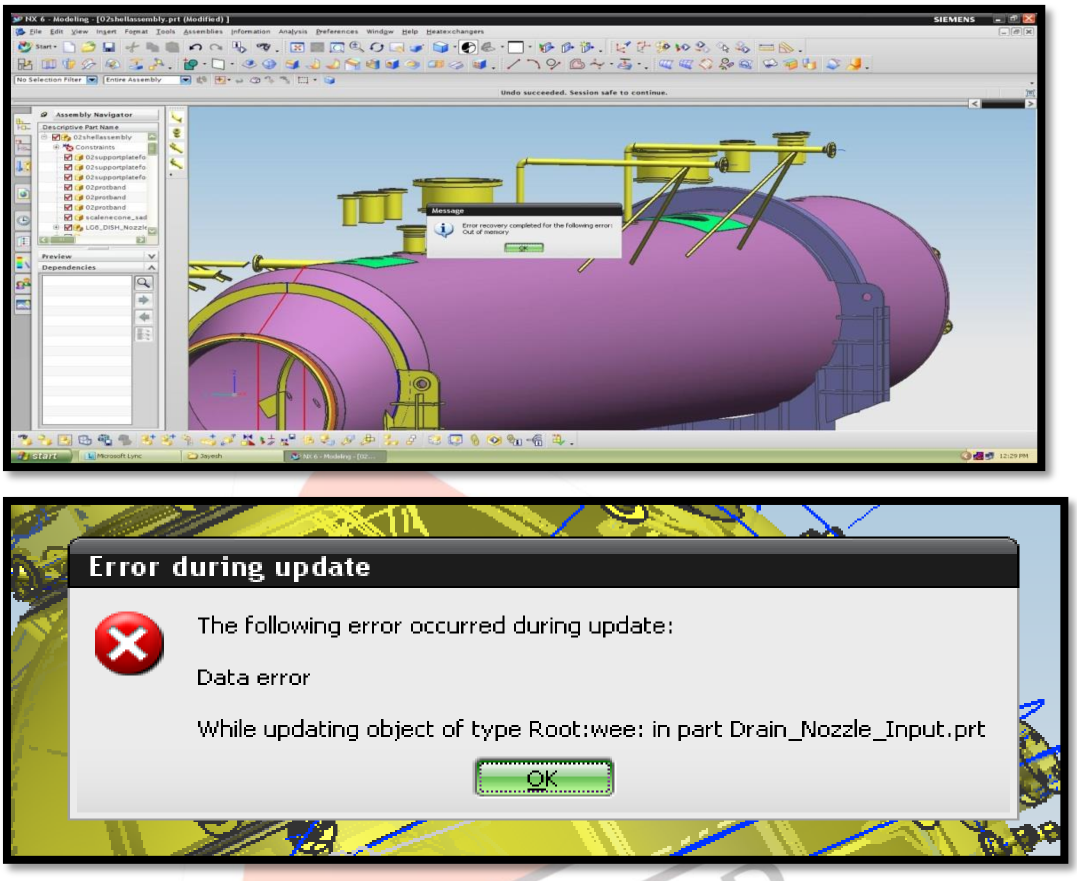

1.START
2.PROVIDE DFA LINKS
3.SET WORKING DIRECTORY
4.OPEN KNOWLDGE FILE
5.IMPORT INPUT PARAMETERS
6.UPDATE , SAVE , CLOSE KNOWLEDGE FILE
7. REPEAT STEP 4.
8. OPEN PART FILE
9.UPDATE PART FILE
10. SAVE PART FILE
11.SAVE AND CLOSE ALL FILES
12.REPEAT STEP 7 TO 11 FOR ALL PART FILE.
13. CHANGE THE SETTING FOR ASSEMBLY FILE
14. REPEAT STEP 4
15.OPEN ASSEMBLY FILE
16.OPEN SUB-ASSEMBLIES ONE BY ONE
17.SAVE ALL THE FILES.
18.SAVE AND CLOSE ALL FILES.
19.DRAFTING
20.END

Fig 1.   Flow Chart for Existing procedure

Doing the things manually following problems are raised:-

1. Single input error result in the fatal error.

2. Time required to open, update and close the assembly files is increased in proportion to the file size. Larger the file more the time is required.
3. High Graphics Power required to the display and processing models.
4. Repetitive task like modifying the parameters become complicated and Time consuming.
5. Sometime the part files may get corrupted, resulting in errors.
6. Repetition work or trial of design required more time.
7. Errors like Memory Access Violation, Out of Memory etc. may affect the system or workstation.
8. The problems are increased in proportion to the number of assemblies, sub-assemblies, parts.


NX 6 - Modeling - [02shellassembly.prt (Modified)]
SIEMENS
Assembly Navigator
Undo succeeded. Session safe to continue.

Message

Error recovery completed for the following error:
Out of memory

OK

Error during update

The following error occurred during update:

Data error

While updating object of type Root:wee: in part Drain_Nozzle_Input.prt

OK

Fig 2. Errors during Update

Because of the mentioned problems the design and production related activities get affected which would cost to the company. So this is one of the major and important part of the product life cycle.

The best way of reducing human interaction and time consumption is doing things automatically. NX Unigraphics can be customized as per the requirements with the help of various tools of customization. NX programming and customization software tools help companies extend and tailor NX solution capabilities to their specific needs. Supporting the knowledge-driven automation capabilities of NX, these tools can be employed by product development personnel and application developers.

II. Customization Approach

The NX Unigraphics has various ways of Automations eg. Knowledge Fusion, GRIP programming, Macros, NX Open, journals, User Defined Objects etc. One of the easiest and economical ways is NX Open using Microsoft Visual Studio. NX offers an API for use with Microsoft's .NET framework. Built on the Common API, this interface provides programmatic access to NX core application functionality, making it possible to create advanced automation programs using any of the .NET-compliant languages, including Visual Basic .NET and C#. Because the API is built on the .NET framework, users can take full advantage of all the benefits provided by that framework.

For earlier mentioned procedure we planned as below

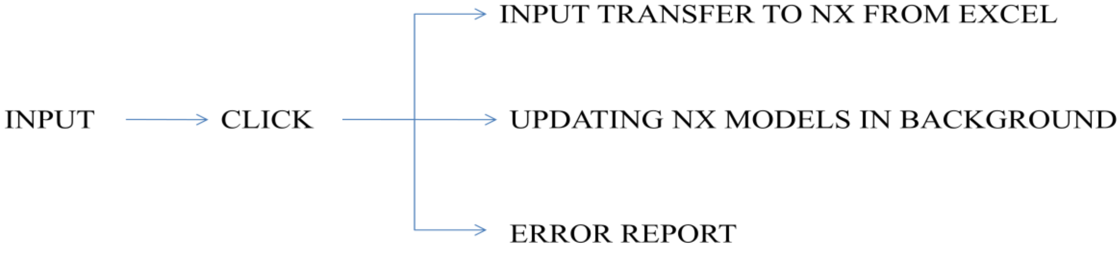
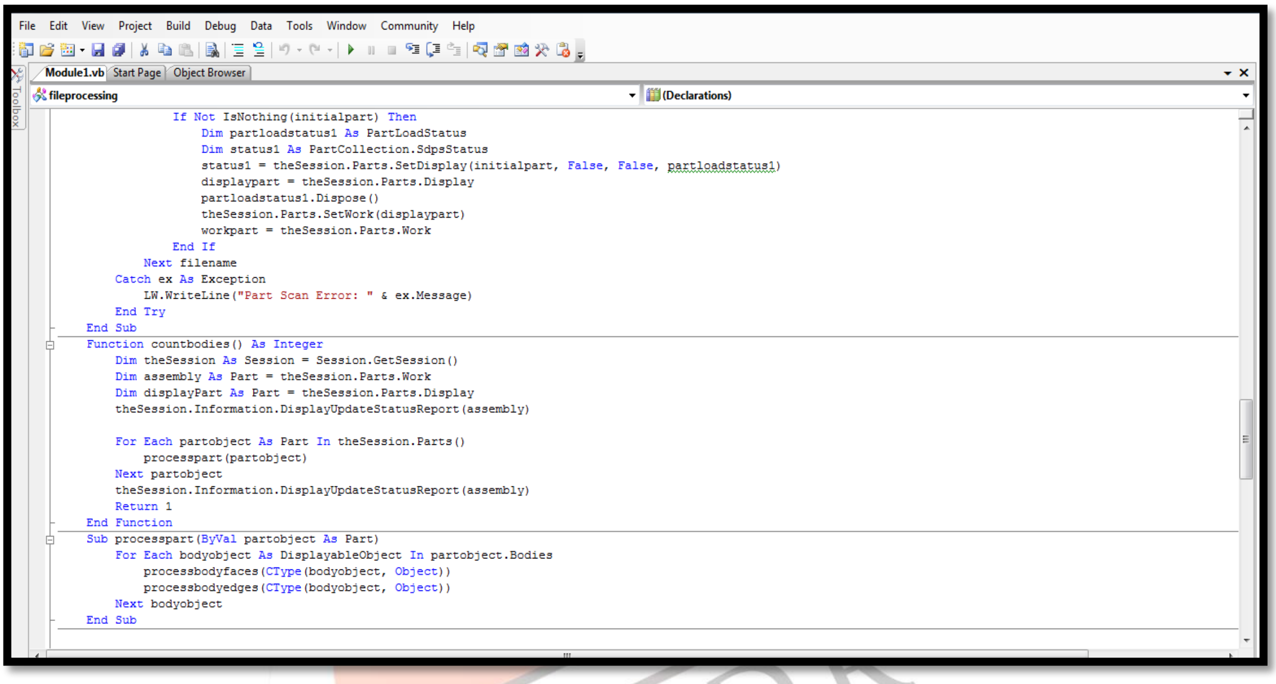

INPUT
CLICK
INPUT TRANSFER TO NX FROM EXCEL
UPDATING NX MODELS IN BACKGROUND
ERROR REPORT

Fig 3. General Plan

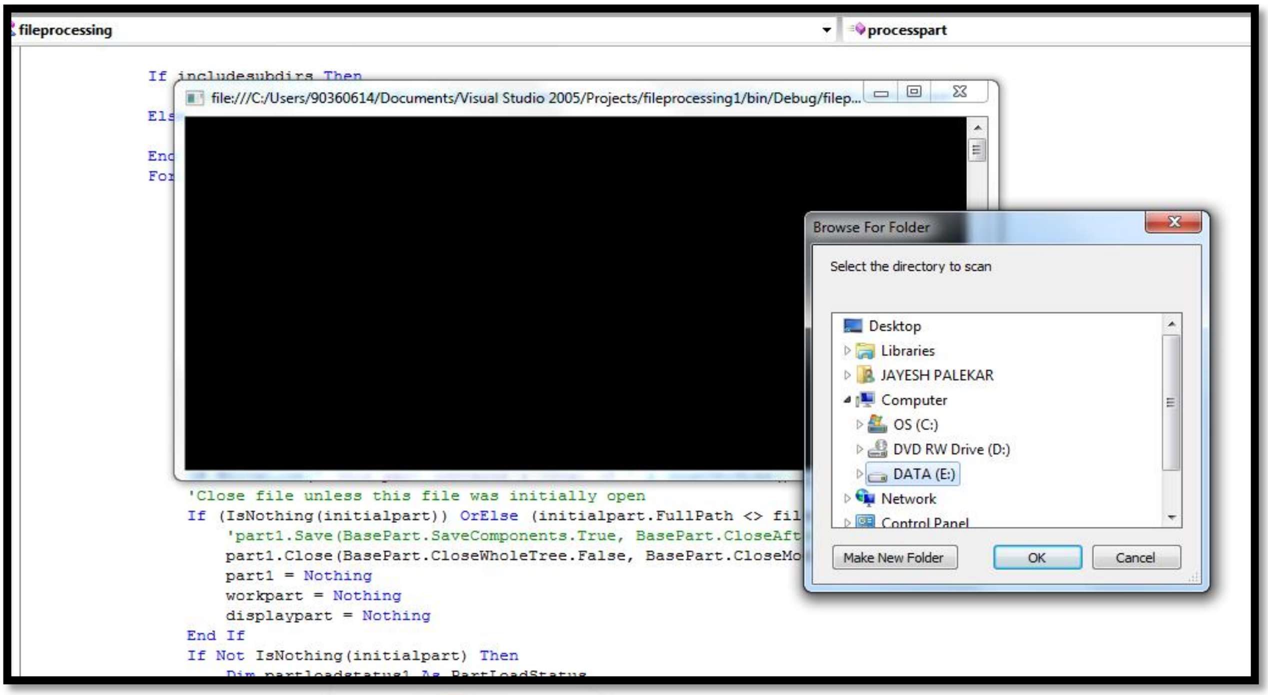
We proposed one API (Application Programing Interface) application with the help of the Microsoft Visual Studio. Within a single click application accept the user inputs (dimensions, parameters, etc.). Then these inputs transferred to the expression file, models updated in the background and the errors generated during process compiled in the error report. Through error report we can identify which file has error or which file which part is failed. Later that part can be modified individually.

For programing purpose we used VB.Net language. To get idea about the basic functioning and programing of operation that has to be atomized, we used the 'Journal' tool in the unigraphics. The Journal capability is a rapid automation tool that records, edits and replays interactive NX sessions. It produced a scripted file from an interactive session of NX which can be run at a later time to replay the session. These sessions can be edited and enhanced with simple programming constructs and user interface components to produce a rapidly generated customized program. Then all edited journals compiled to form a single program. These edited journals were relating to set the working directories, to provide the required DFA links and data files, to open, save and close the part files. For updation of a part file logic developed, which process each and every part of the assembly, body, surface, edge in sequence. To record the geometrical error generated while updating the part file used listingwindow class which record the strings and text messages.



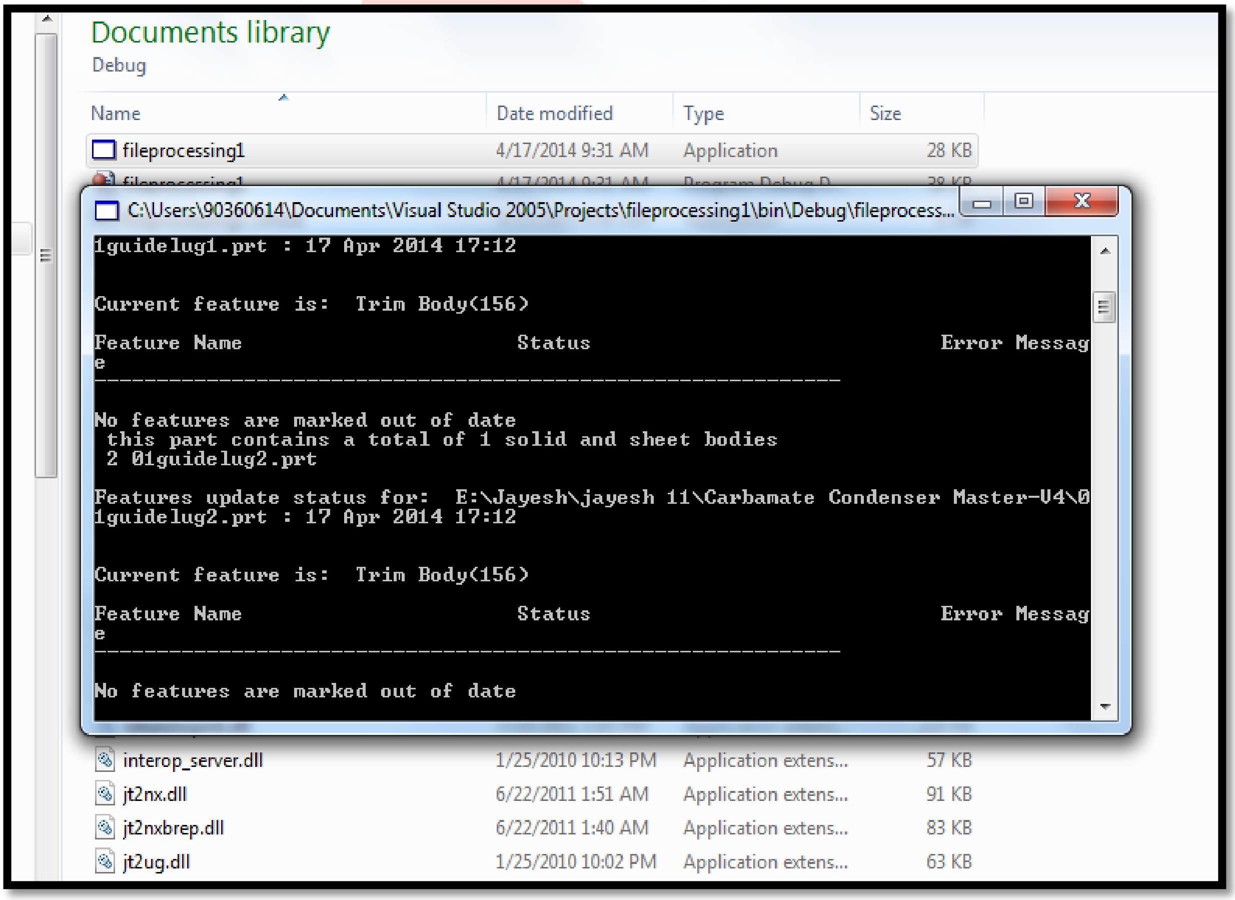
```
                If Not IsNothing(initialpart) Then
                    Dim partloadstatus1 As PartLoadStatus
                    Dim status1 As PartCollection.SdpsStatus
                    status1 = theSession.Parts.SetDisplay(initialpart, False, False, partloadstatus1)
                    displaypart = theSession.Parts.Display
                    partloadstatus1.Dispose()
                    theSession.Parts.SetWork(displaypart)
                    workpart = theSession.Parts.Work
                End If
            Next filename
        Catch ex As Exception
            LW.WriteLine("Part Scan Error: " & ex.Message)
        End Try
    End Sub
    Function countbodies() As Integer
        Dim theSession As Session = Session.GetSession()
        Dim assembly As Part = theSession.Parts.Work
        Dim displayPart As Part = theSession.Parts.Display
        theSession.Information.DisplayUpdateStatusReport(assembly)

        For Each partobject As Part In theSession.Parts()
            processpart(partobject)
        Next partobject
        theSession.Information.DisplayUpdateStatusReport(assembly)
        Return 1
    End Function
    Sub processpart(ByVal partobject As Part)
        For Each bodyobject As DisplayableObject In partobject.Bodies
            processbodyfaces(CType(bodyobject, Object))
            processbodyedges(CType(bodyobject, Object))
        Next bodyobject
    End Sub
```

Fig 4. Programing Window



```
        If includesubdirs Then
        Els
        End
        For
            'Close file unless this file was initially open
            If (IsNothing(initialpart)) OrElse (initialpart.FullPath <> fil
                'part1.Save(BasePart.SaveComponents.True, BasePart.CloseAft
                part1.Close(BasePart.CloseWholeTree.False, BasePart.CloseMo
                part1 = Nothing
                workpart = Nothing
                displaypart = Nothing
            End If
            If Not IsNothing(initialpart) Then
                Dim partloadstatus1 As PartLoadStatus
```

Fig 5. Run Time mode:- Selection of the working directory location.


Documents library
Debug
C:\Users\90360614\Documents\Visual Studio 2005\Projects\fileprocessing1\bin\Debug\fileprocess...

Name	Date modified	Type	Size
fileprocessing1	4/17/2014 9:31 AM	Application	28 KB
fileprocessing1	4/17/2014 9:31 AM	Program Debug D...	28 KB
interop_server.dll	1/25/2010 10:13 PM	Application extens...	57 KB
jt2nx.dll	6/22/2011 1:51 AM	Application extens...	91 KB
jt2nxbrep.dll	6/22/2011 1:40 AM	Application extens...	83 KB
jt2ug.dll	1/25/2010 10:02 PM	Application extens...	63 KB

```
1guidelug1.prt : 17 Apr 2014 17:12


Current feature is:  Trim Body(156)

Feature Name                      Status                           Error Messag
e
------------------------------------------------------------

No features are marked out of date
 this part contains a total of 1 solid and sheet bodies
 2 01guidelug2.prt

Features update status for:  E:\Jayesh\jayesh 11\Carbamate Condenser Master-V4\0
1guidelug2.prt : 17 Apr 2014 17:12


Current feature is:  Trim Body(156)

Feature Name                      Status                           Error Messag
e
------------------------------------------------------------

No features are marked out of date
```

Fig 6. Run Time mode: - Recording of errors generation (Output File)

The following table will illustrate the comparison between existing manual procedure and the proposed automated procedure.

PROCESS	EXISTING	PROPOSED
1. Excel Input	Manual	Manual
2. Transfer of Excel Data to Text	Manual	Manual
3. Setting working directory and providing required library links	Manual	Automatic
4. Importing Excel Input	Manual	Automatic
5. Opening Model Files	Manual	Automatic
6. Updating Model Files	Manual	Automatic
7. Error Report Generation	Automatic (Individual)	Automatic (Combined)
8. Saving Model Files	Manual	Automatic
9. Closing Model Files	Manual	Automatic
10. Drafting	Manual	Manual

Table1- Comparison of existing and proposed system

III. Result

Due to the automation we get following benefits:-

1. Time required for the operation is reduced tremendously. For one random part file size of 8MB having 3 sub-assemblies, 45 parts and 180 features, we got the following time analysis.

PROCESS	EXISTING SYSTEM TIME(Sec)	PROPOSED SYSTEM TIME(Sec)
1. Setting working directory and providing required library links	60	Within fraction of sec
2. Importing Excel Input	30	Within fraction of sec
3. Opening Model File	30	10
4. Updating Model File	300	60
5. Saving Model File	120	25
6. Closing Model File	30	10
Total	**570**	**105**

Table2- Comparison of existing and proposed system (Time Analysis)

Approximately 75% to 80% man hours are saved.

2. Operations are generated in the background (in MS dos mode), no need of opening the unigraphics environment hence no need of the graphics, so the operations are quicker and graphics power is saved. Other computer operation can be performed simultaneously because of the less RAM is required for the operation.
3. All the errors are compiled together, which can be retrieved in later stage.
4. Large part files can be handled easily.
5. Repetitive design tasks can be easily performed.
6. The error generation due to graphics consumption (like Out of Memory) is less.

As the system is beneficial, we came across some issues on which we are working on it.

1. To improve the quality of the updation logic like some parts have to update twice thrice to give the desired result. Rarely the application doesn’t give the required result.

2. To make the set up file of the application which can be used on all workstations where no need of unigraphics software or having less graphic powers.

IV. CONCLUSION

We tested the application for various inputs parameters and designing conditions and came across the following conclusion, how the customization approach is beneficial to the designing department of the company,

1. As time saved for the operation the Product Lifecycle Time is reduced.
2. Drawing generation tasks from parametric models become efficient.
3. Designing cost to the company is reduced.
4. Percentage of the error generation is reduced.

V. REFERENCES

[1] Ming C. Leu, Akul Joshi, "NX5 FOR ENGINEERING DESIGN", Missouri University of Science and Technology Rolla, Missouri 65409

[2] Randy H. Shih, "Parametric Modeling with UGS NX5", Oregon Institute of Technology

[3] "OPEN API STUDENT GUIDE", Feb-2004, MT13110-Unigraphics NX2

[4] "NX V6.0 Help Guide and Release Notes", Siemens PLM

[5] MSDN Help Guide

[6] SIEMENS – UGS PLM Software, http://www.plm.automation.siemens.com/en_us/

[7] NX Journaling- http://www.nxjournaling.com/

[8] Siemens: UG/NX Forum- http://www.eng-tips.com/threadminder.cfm?pid=561

[9] LinkedIn-NX Open .NET a subgroup of NX Users (Formerly Unigraphics) Siemens PLM- http://www.linkedin.com/groups?gid=2467809