International Engineering Research Journal Page No 34-41

Development of Software for Converting Non-Native CAD Data While Working with Heterogeneous CAD Data using Creo (May 2016)

Jude A. Rodrigues[1], *M.E Student*, Prof. A. R. Suryavanshi[2], *HOD, Mechanical Department (ZCOER)*

Abstract— **In large scale design process, designers generally come across a complex situation where variety of Computer Aided Design (CAD) tools are used for developing a 3D model and assemble the same within the product assembly. As a solution to deal with such situations, advancements in these modelling applications were made in order to support 3D data coming from different commercial CAD tools. This mechanism of handling the heterogeneous CAD data was based on the macro parametric approach. The approach of working with macro parametric file for building a heterogeneous structure holds strong until users do not find the need of modifying the files coming from other modelling applications. This limitation arises due to the diverse nature of the geometry building algorithms used in all the commercial CAD software's. In order to make data handling simpler and robust, there is a need to convert the files undergoing a design change into the native application format without converting the other set of files which are not undergoing design changes within the heterogeneous CAD structure. At present, converting the non-native models takes considerable time of the user. This paper implements an approach of converting the non-native models much faster without the user actually needing to spent much time on the non-design activities.**

Index Terms— **CAD Data translation, Heterogeneous CAD, Multi CAD.**

I. INTRODUCTION

The phase of product development is rapidly transforming based on the basic element of information technology. The market now demands more and more complex products to be collaboratively developed by multiple departments or groups within a company spread across different geographical locations. Even, in many cases, outsourcing the design activities has proved to be the most cost efficient technique in product development. Even the usage of modelling software's has drastically changed the pace at which the product is developed over its lifecycle. There are some commercial software's in the market that provides good 3D modelling capabilities. Every software has a unique feature of its own. Based on the need of the designer or the application engineer using the software, the modelling software is selected so that it helps to develop the product rapidly.

With the choice of using multiple designing software's, most of the manufacturing companies lend up in creating 3D data related to a single product in multiple designing software's. This variation in data formats brought the need of implementing multi-CAD interoperability which is also known as heterogeneous CAD data. In Heterogeneous CAD data management, all the data coming from different modelling tools is managed under a single tool. With the advancement in these modelling software's to support the heterogeneous data effectively, the user can use the data coming from a different designing tools (also known as non-native CAD data) without the actual need of converting it into the native data format. This is achieved by using the Extended Markup Language (XML) file based mechanism known as macro-parametric approach. This approach serves as the best fit solution to the cases where the designer does not intend to create a separate file for the existing 3D model serving only single design intent.

In macro-parametric approach, the geometric data as well as the metadata is exchanged among the different modelling applications using a neutral XML. All set of modelling commands for every designing tools is studied in details and on this basis, a neutral set of commands are created in order to interpret all the modelling information related to every commercial modelling tool efficiently. This all information is stored into a temporary macro file. After the user explicitly saves the design model, this temporary macro file in converted into an XML file using a data translator. This workflow is depicted correctly in fig. 1.

The macro parametric approach is worth until there are no design changes that need to be made on the non-native file by the designing tool. Hence, this approach is usually implemented only when the user needs to visualize the entire product information using a single modelling tool. Even considering the mechanism of over-writing the macro file created by one modelling application is a high risk factor. This approach will not guarantee the persistency of the geometric information. The various flaws of using the macro-parametric

"This work is supported by Parametric Technology's Cooperation Inc. India Pvt. Ltd." Jude A. Rodrigues is a student pursuing M.E. from ZCOER, Pune 411 041 (e-mail: jude.rodrigues25@live.com)

approach are:

1. The internal terminologies for modelling commands that are used in every modelling system can change over every new release of the CAD application. This will then need an update to the corresponding Application Program Interfaces (API) for every modelling command. Doing this as a maintenance activity is very expensive.

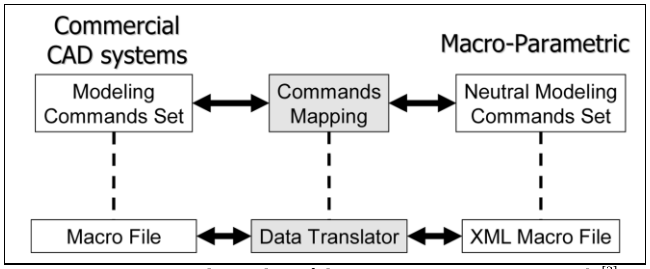

Figure 1. Mapping relationship of the Macro-Parametric Approach [2]

2. The coordinate systems of different software's are not the same. Some use global coordinate system as the base and local coordinate system as reference while creating a sketch. This kind of variation becomes difficult to be handled by the translator. Unifying this can cause problems if the other CAD system uses local co-ordinate system globally. The translated data will be misleading and thus leading to the phase where the geometric data is corrupted.

3. Some of the other well-known problems of data exchange is related to naming the topology in a persistent way. While building a geometric model, there are several ways of generating the model entities and assigning an external reference ID. But, during the data exchange process, these entities can be represented incorrectly due to the varying algorithms for the entity as well as ID generation. This may cause unhandled errors during the data exchange.

4. Every CAD applications handle the requirement for accuracy differently and have some specific definition of the same uniquely. Some may be using Relative accuracy as default where as others may be using absolute accuracy. This variation in accuracy may not support the parametric exchange of the data efficiently. This happens because we use different algorithms as well as representations for every CAD system.

II. PROBLEM STATEMENT

With these many issues in the macro-parametric approach, it seems that the only way to carve out a stable 3D model is to use the native data structure of the modelling system used for storing all the information from the macro file. The creation of this native CAD data file should be valid only if there is a design change being done on the non-native 3D models. This check should be put in place because the user should not lend up converting the files that are not undergoing a design change.

This paper will demonstrate an approach to convert the non-native CAD data into the native CAD data file by reading the macro file which is already created by the data translators. This mechanism will only be implemented if the user is working on a heterogeneous assembly in the modelling session. Also, an approach of mapping the topological entities is discussed so that the geometric constraints are not manipulated during the data conversion activity.

By the implementation of this mechanism, the designer is not expected to invest much of the time in performing non design activities like converting the non-native CAD data into native data and then replacing the converted file into the structure of the heterogeneous assembly. This in-turn can save a lot of designer's time enhancing them to invest more in the design activities.

III. LITERTURE REVIEW

Though all modelling applications mostly focus on geometric modeling and providing an extensive support for detailed designing, they have advanced out to support complex product design. Modelling applications providing the mechanism of top-down product designing was found out in 1990s by Mantyla [1]. This research was mainly focused on supporting early design stage by enabling the inheritance of information by generating a consistent relationship between different stages of design. The basic needs of any design system to support top-down design of a product were studied. Also, a prototype system was developed that demonstrated the mechanism for supporting a top-down design approach.

Although this was a preliminary work lacked in details to develop an application supporting the top-down design approach, it pointed out an important direction for CAD research. Based on this, computer supported sketching method was adopted to support concept design by Y. Zeng [2]. This helped the designers to construct 2D sketches as different entities and then later use these sketches as reference for constructing a 3D object. This kind of modelling was known as entity based modelling.

As a most reliable way of geometric modelling, feature based modelling approach was an advancement done by Csabai et al. [3] in CAD technology. This method enabled the inheritance of information between different stages of design. In order to fill the gap between the conceptual design phase and detailed design phase, the implementation of design spaces as well as functional features was introduced in 3D layout module. This module provided an extended support to the top-down design process.

Modelling application like Pro/E has also implemented an approach to support top-down design process by making the use of skeleton model as a source to save the design information for the assembly. This design information includes the dimensioning between parts as well as the geometric constraints used to define the position in of every part within the assembly [4].

This extensive support for top-down design approach has induced many users to work in the assembly context itself while trying to modify the design intent of the part under the assembly. Using this approach enables the user to see the changes reflected in the assembly and guides the user to finalize upon the design intent looking at the related interference.

Guk-Heon Choi et. al. [5] proposed a mechanism of exchanging the 3D CAD data among different commercial modelling application by storing the sequence of modelling commands in a file known as macro file. But this research was more biased on the part modeling approach that failed to address the assembly modelling support.

Another research by Jinggao Li et. al. [6] studied the different errors found in using the macro parametric approach to develop a set of standard modelling commands. In general, all the topology mapping differences, problems related to varying co-ordinate system mechanism as well as the accuracy was discussed.

The work by W. Li et al. [7], describes that collaborative designing can be divided into two types: (1) Visualization-based collaborative CAD; and (2) Co-design oriented collaborative CAD. The visualization-based collaborative CAD mainly focusses on the mechanism of supporting the visualization of the entire product outside the modelling tool and allows the user to perform preliminary design checks such as clash detection, tolerance check, mashup creation etc. Whereas, Co-design oriented collaborative CAD comes up with ability generate a geometric model and modify the same across different modelling applications.

Our main focus is to study the work on the approach of developing an assembly modelling system in a collaborative design phase which mainly focusses on Co-design oriented collaborative CAD mentioned by W. Li et. al. [7]. Further, Mori and Cutkosky [8] studied the process of agent-based collaborative assembly design. In their architecture, each part or a component of an assembly is managed by a single design with others based on the theory of Pareto optimality. Here, the way the design agents communicate with each other is simple and limited; it seems difficult for the framework to support more complex collaborative activities.

Chen et al. [9] found a way to collaboratively work in a modelling system known as E-Assembly. The E-Assembly represents a Masters Assembly Model (MAM) which is completely web based as well as a Slave Assembly Model (SAM) which generally focus on the client. In Master Assembly Model, as the name suggest, the complete representation of the assembly is stored. Where as in the Slave Assembly Model, only the information is stored. The SAM is generally meant to work on client in cases where the user needs to view the visualization as well as to manipulate the structure on the client. E-Assembly general targets the scenarios where the collaborative assembly modeling between Original Equipment Manufacturers (OEM) and suppliers is considered. With the help of this mechanism, multiple users can work collaboratively in order to build up an assembly model by fetching the components from the web and store the entre representation as well as the structure on the web. The main drawback of this system is that it does not thoroughly support the top-down assembly design process.

The collaborative product design can be split into two categories: Collaborative component design and Collaborative assembly design [10]. The case where an OEM develops an assembly of the final product by referring parts or components that are designed by part suppliers who use a different modelling application, falls under the category of collaborative assembly design. In the collaborative component design environment, users or designers actually get the data (parts or components) from their suppliers who use a different modelling application, and then convert this data into a neutral file format. After the conversion, this neutral data is entered into a central virtual repository which can be referred by any designer working on building the product [11-14].

Jinsang Hwang and Duhwan Mun [15] have proposed a new method of using a Neutral Reference Model (NRM). This method describes the way of sharing and propagating Engineering Change (EC) information among OEM’s and part suppliers working in a distributive collaboration environment. Here, the concept of skeleton-based change propagation and its process are discussed and a mechanism has been defined. An EC Propagator, a prototype system, represents EC information intuitively and precisely for designers, since a NRM is a designer-friendly media. Using the NRM without necessitating reference to parametric CAD data of other components designed by other collaborating companies makes it possible for the designers to work on the same without investing much effort in design parameterization.

Ari Rappoport [16], effectively communicated the main concepts behind the architecture of universal CAD data exchange. This data exchange mechanism uses a star architecture with a central Universal Product Representation kernel that contains data structures for representing the union of the data and operation types supported by modelling systems on that specific data. For a system to work on this architecture, two modules need to be implemented: export and import. The function of the export module is to translate the data from the designing application using the appropriate conversion mechanism depending on the type of data exchange format. Whereas the import module is responsible for actually converting the data coming from other modelling systems into the native CAD application format. For each of the data type, the algorithm actually detects the incoming data format and takes then invokes the necessary data translator to carry out the data conversion activity. In case of any errors, the error is read and then with a corrective action, the translators are executed again until the data conversion activity is not taken to completion.

In another approach of transferring the CAD data into a lightweight XML file, In-Ho Song et. Al. [17] described a way of extracting the geometric information within an assembly file and then stores it in the XML file. This XML file is later referenced in the PDM system and the geometry is constructed on the basis of the extracted information.

Tae-Sul Seo et. Al. [18] introduced an ontology-based method to enable the semantic interoperability of the feature-based design data. The introduction of this interoperability allows the modelling systems to share and exchange CAD models build up using the feature-based functions. STEP AP224 was analyzed to find out the relational hierarchy of modeling features and based on this information; shared feature ontology was constructed along with the feature-resource and the macro-parametric approaches. In macro-parametric approach, in order to enable the interoperability, some relational axioms have been defined. As a part of evaluation, the design features of the two modelling applications have been fetched and edited with the help of a user interface which behaves like an application supporting both the modelling applications. But using this approach of referring the design features of the commercial modelling application becomes a bigger problem when the two modelling systems try to compete with one another in the market. Any future support to exposure of the API's may get changed or even removed at times thus forcing the CAD development team to come up with a new release that encapsulates the changes. In order to bridge these gaps and enable the user to make the design modifications using a single system, the following mechanism has been implemented.

IV. METHODOLOGY

A. *Extracting data from Heterogeneous CAD file*

The major idea in this section is to read the information stored in the macro file generated by the CAD translator and extract all the needed information from it so that we can push them into the native file format. So, while doing so, the major intent is to understand the hierarchical ordering of components or parts used within the assembly structure along with their respective constraints with reference to one another [19].

The support for Heterogeneous CAD data to be handled under a single data file was implemented by PTC Inc. in its CAD software Creo 3.0. In most of the cases, the macro files which stores the translated information is not in encrypted format. But since, Creo being a commercial product and the data integrity being the key concerns for the people using Creo, the macro file is put in an encrypted format which is recognized as .creo file.

Whenever we retrieve the non-native CAD data in Creo, the translator embedded within Creo, checks the authoring application of the file being opened in Creo and based on that, the respective CAD data translator is invoked. These translators work on the architecture of STEP AP224 [20]. The example of a non-encrypted macro file is as shown in fig. 2.

```
<block base shape >
 <!--  Feature Definition -->
 < id="feat1">
 <length>40</length>
 <height>50</height>
 <width>13.4</width>
 <removal_boundary = "false">
 <vee_profile>
 <profile_radius>0</profile_radius>
 <tilt_angle>0</tilt_angle>
 <profile_angle>90</profile_angle>
 </vee_profile>
 </>
 <its_tolerance>
0.05
 </its_tolerance>
 <its property>
 </feature>
 <feature>
 <placement>
 <location x="120" y="195" z="125"/>
 <axes>
 <x_axis x_direction_ratio="1" y_direction_ratio="0" z_direction_ratio="0"/>
 <y_axis x_direction_ratio="0" y_direction_ratio="1" z_direction_ratio="0"/>
 <z_axis x_direction_ratio="0" y_direction_ratio="0" z_direction_ratio="1"/>
 </axes>
 </placement>
```

Figure 2. Sample macro file

Since we are dealing with an encrypted macro file, there is no way to extract the information from the macro file directly. Hence, the only way of getting the needed information from the macro file is by the means of using the API's used to read the macro file. When the non-native 3D model is opened in the modelling application, the macro file is read and the needed information is translated appropriately into geometric features as well as the metadata models which is understood by Creo. The API's that are used to read the macro file are directly called in the set of code that will be responsible to put all the information available in the modelling session to the data structure of the part file. The information generally consist of all the elements such as the model tree, the datum system, the skeleton model, the Constructive Solid Geometry (CSG) information as well as bounding box information related to every part or component being assembled under it can be referred as the Heterogeneous CAD assembly. Another important aspect while working in 3D modelling environment is the topology. The macro file does not contain much information related to the topology. The references needed to define the topology of the component are generated in the modeling session. Hence, getting this information from the session is also a very important.

The way in which the macro file is read is important in order to know the way in which the information read is by Creo will be stored back. So, while reading the macro file, each header of the macro file is read sequentially and the respective API of the feature is called in order to construct the geometry as well as the graphics. This helps us to run the feature creation wizard in the background to generate the needed geometry as well as the graphics for visualization.

The first header: 'id' helps in giving the boundary information about the geometry. The second header i.e. 'vee_profile' is for storing the spline information. The next 'its_tolerance' is for specifying the geometric tolerance in case of modelling tools that support Geometric Dimensioning and Tolerancing. The next header consists of 'its property' that constitutes the custom parameters added into the 3D model, e.g. Material, Modelled By, Date etc. The last header, 'placement' is for holding the location of the part in the heterogeneous assembly.

Each of the headers mentioned here will be mapped to the respective API's. This mapping will create a stable background for feature creation. After the creation of the features, the information will be translated to data structure of the native file format. This translation takes place for the non-native CAD file when the user initiate a 'Save' for a specified part undergoing design changes.

B. *Conversion Mechanism*

This mechanism demonstrates the working of the conversion algorithm of the macro file into a file recognized by a modelling tool. The user working on the heterogeneous CAD assembly in a top-down design approach when selects the non-native part or component for modification (feature addition), is provided with an option to automatically convert the non-native file into native CAD file. At the same time, another option of not to convert the non-native document is facilitated in order to give the user the flexibility to try design approximations on the selected part or component without actually having the need to save the model as shown in fig. 3.

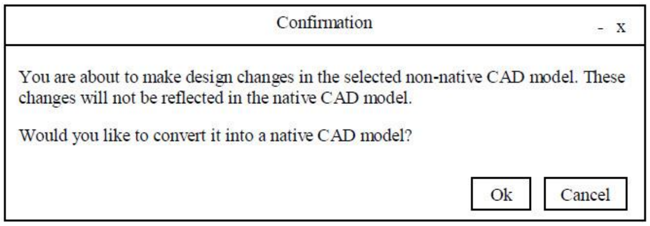

Figure 3. Convert Dialogue Box

The selection of ‘Ok’ will initiate the conversion of the non-native document into the native CAD document. ‘Cancel’ will allow the user to proceed with the design changes on the non-converted file i.e. macro file.

If the user selects ‘Ok’, then all the information stored in the macro file of the selected document will be translated into the native file. The work of interpretation is done by the macro file translator. This process runs in the background and is responsible for writing the data in the macro file and then save it into the native file format. This will provide a stable platform for the geometric features to be written back in the CAD files data structure. This kind of conversion on the fly will never need the CAD data translators to run in the background as all the information will be fetched from the macro file thus eliminating the overhead cost of using the translators for data conversion at the users end. Even the process memory of the computer in use will be less consumed thus providing a dedicated memory for the modelling tool to actually render the heavy data in session effectively.

This conversion process follows the bottom up conversion approach. i.e. if we select the non-native component to the lowest level in the assembly structure for conversion, then the immediate parent is checked if it is a native CAD assembly. If not, then the parent is also considered for conversion. This is done unless the parent is not found to be a native assembly.

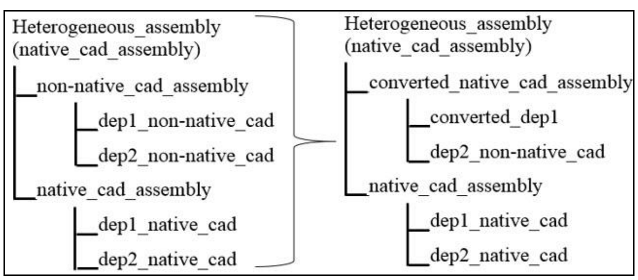

Figure 4. Non-native CAD models to native CAD models

In the above example, the user had selected ‘dep1_non-native_cad’ for conversion. The algorithm detected that the immediate parent: ‘non-native_cad_assembly’ is also not a native CAD assembly; hence it was selected for conversion. The selection algorithm stopped as soon as it detected that the parent of ‘non-native_cad_assembly’ is a native CAD assembly.

C. Mapping the topological references

During the retrieval of any CAD data, the CAD tool is responsible for generating a unique ID for every component or part known as Component ID. Every CAD tool has its own mechanism of generating the component ID’s in a consistent way. Even the topology data is also created efficiently in every CAD tool today. In order to have a stable mechanism of mapping the topological information, it becomes necessary to use both the component ID’s as well as the topological names defined for a CAD component to generate the mapping references. In order to use this persistent information, there arises a need to extract this information from the session of the CAD tool, so that we can use both the information to generate an algorithm for mapping. For this purpose a new function is needed to be written. This function plays the role of extracting only the topological names and the component ID of the component and then makes it available for reference in another function where we carry out the mapping of the topological information present in the old non-native CAD part with the topology of the newly created part.

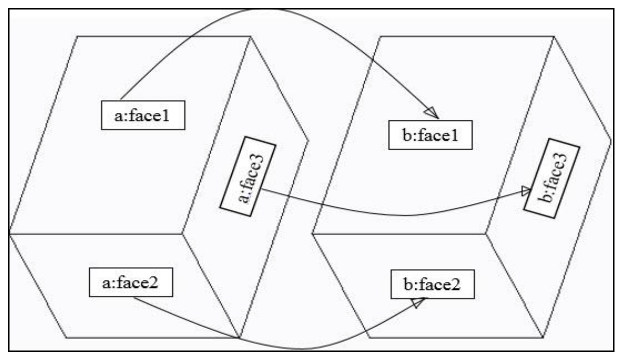

Figure 5. Topology Mapping

For a single component retrieved in the CAD tool, the topological reference is defined in the following manner:

<component_id>:<topology_reference_name><id> …1

In case of an assembly component, the topology references are generated as follows:

<assembly_id>:<component_id>:<topology_reference_name><id> …2

During conversion, the topological information within the old model needs to be transmitted into the new model. This work is done by the macro file translator. The references are then mapped based on the following equation:

<new_component_id>:<topology_reference_name><id> = <component_id>:<topology_reference_name><id>

This mapping mechanism is carried out by a topology bus which is capable of pertaining the old reference information along with the new reference information.

After the successful mapping, the newly created data file is ready to be saved in order to inherit a stable data structure of the native design application. On completion of the conversion activity, the user is permitted to precede ahead with the design modifications (feature additions). The newly added feature are then written back to the converted macro file in the same way the modelling tools writes in its own native file.

V. IMPLEMENTATON AND CASE STUDY

The macro files conversion mechanism while working within a heterogeneous assembly was implemented and tested in a modelling tool owned by PTC Inc. known as Creo. PTC Creo provides various API's for the end users to automate their design by using the best practices. The CAD tool is built up using C++ language. Hence, we used C++ to develop and implement this mechanism.

A. Converting the non-native CAD data

The first steps were creating the heterogeneous CAD structure. We used the data created in Solidworks. A Creo assembly was created and then the Solidworks data was assembled under the Creo assembly (which can now be termed as the heterogeneous assembly. Refer fig. A

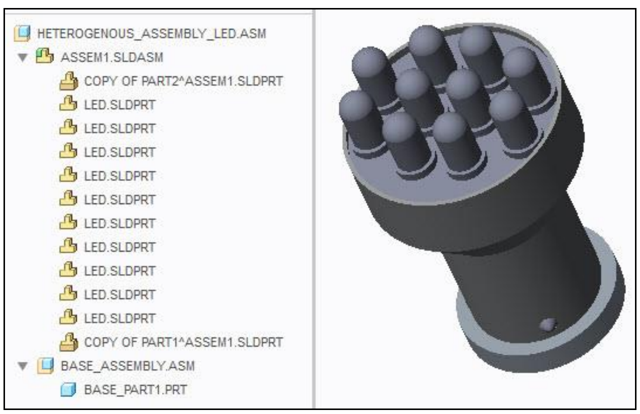

Figure 6. Heterogeneous Assembly

Next, a solid works component was selected and activated in order to make the design changes (Adding a Round feature). The user is prompted to convert the Solidworks model into a Creo model.

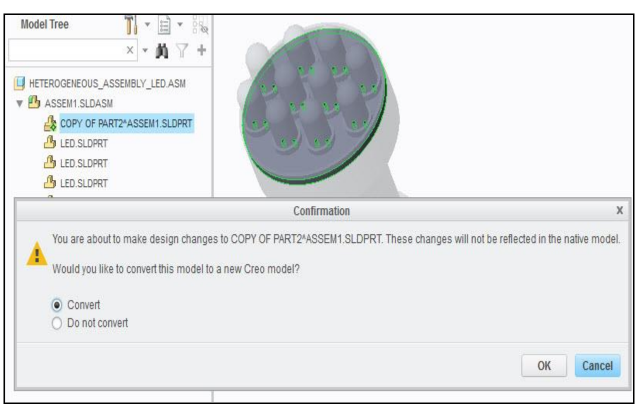

Figure 7. Convert to new Creo model UI

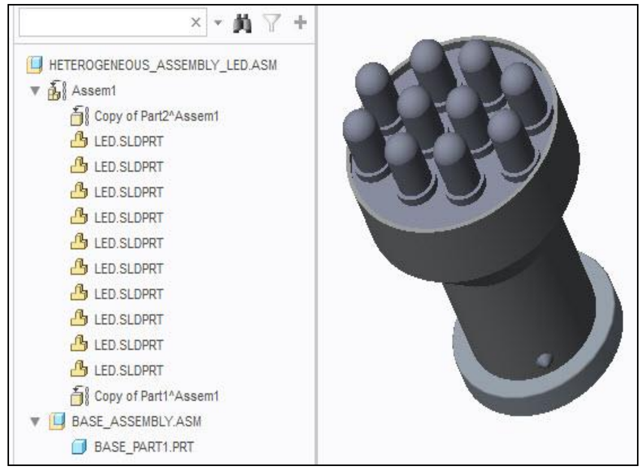

Figure 8. Converted models in the graphics area as well as model tree

Now, here if the convert check box is ticked and the user selects 'Ok', then the selected part 'Copy of Part2^Assem1.SLDPRT' will be converted into a Creo part.

After the successful conversion, the user is allowed to activate the converted component and proceed ahead with the design modification.

B. Choosing not to convert the non-native CAD data

In cases where the user does not want to confirm the changes in the CAD data file, he / she has the flexibility to make design approximations by choosing the option of not converting the CAD data.

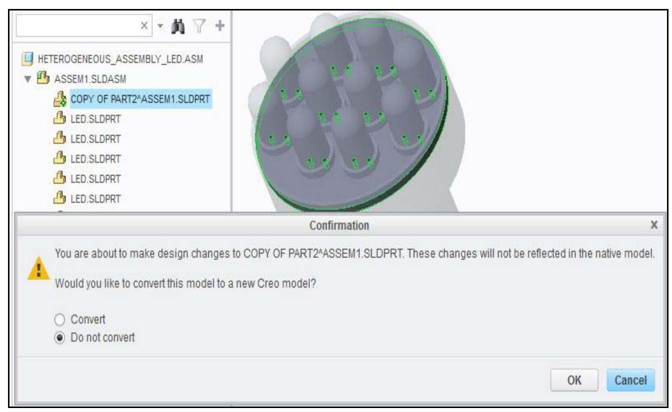

Figure 9. Conversion UI with the user choosing not to convert

On selecting 'Do not convert' and click 'Ok', then user is navigated to the feature creation wizard. Where on the intended approximations are carried on and the changes are reflected accordingly in the graphics area.

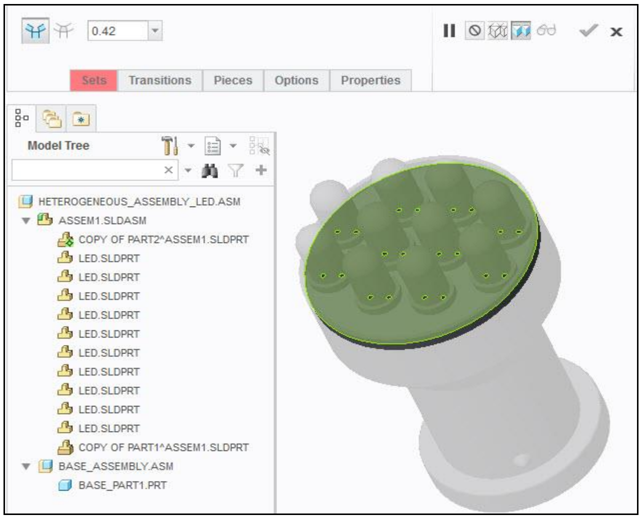

Figure 10. Feature definition on the non-native CAD doc

VI. RESULTS AND DISCUSSIONS

This procedure is expected to work for all the level of dependents within the heterogeneous CAD data structure. The average time for performing the conversion activity without the presence of this method was recorded and was found to be as per the readings.

This implementation was evaluated on the components present at different level within the assembly. The assembly was constructed to have the components up to third level. The readings for the time taken to perform the non-design activities were initially taken without implementing the approach. It was observed that as the level at which the component is present increases; the time needed to perform to conversion and replace activity within the assembly goes on increasing. Refer table 1.

TABLE 1
TIME TAKEN FOR CONVERSION WITHOUT CONVERT ON THE FLY MECHANISM

Sr. No.	Level of component in assembly	Time taken (secs)
1	Single component	12
2	Assembly with one component	15
3	Component at second level	21
4	Component at third level	27

Now with the implementation of this approach, it is observed that the average time needed to perform the conversion and replace action for any component within the assembly is the same at any level within the assembly. Table 2 gives the readings taken for the same levels. The user doing the design action was also having the same expertise who performed the actions without this mechanism being implemented.

Hence, it is expected that this approach will not only save the time needed to perform the design changes, but also will keep the user care-free about the data handling activities like naming conventions to be used while conversion, the reference mapping to be done during replace etc.

TABLE 2
TIME TAKEN FOR CONVERSION WITH CONVERT ON THE FLY MECHANISM

Sr. No.	Level of component in assembly	Time taken (secs)
1	Single component	8
2	Assembly with one component	8
3	Component at second level	8
4	Component at third level	8

VII. CONCLUSION

With the mechanism defined in this paper, the designer is provided with an approach that will automatically carry out the non-design activities like converting the non-native data into native data and then replace the converted file into the assembly structure of the product at any desired level within the assembly. Thus reducing the non-productive time from design perspective.

Calculating the actual non-design time spent in managing the 3D data, it is seen that the percentage saving in time goes on increasing as the complexity of the assembly increases. For a simple part, the conversion time actually saved using this approach is approximately 33%. For an assembly with one component, it is approximately 46%. For an assembly with component at the second level, it is 61%. Even, as the conversion is carried out on the basis of the macro file, the overhead cost of each user using the data translators is also reduced depending on the cost of data translators for respective third party CAD application.

Now, with the introduction of the mechanism of creating a new native document, problem arises when there is an update made on the non-native data using its respective modeling tool. In order to maintain the associativity and enable the user with the ability to fetch the updates from the newly iterated non-native CAD data, the future work relies upon the establishment of such mechanism.

ACKNOWLEDGMENT

I wish to express my sincere thanks and deep sense of gratitude towards my guide and HOD, Prof. A. R. Suryavanshi and P.G. coordinator, Prof. J. A. Shaikh for their guidance and constant encouragement in all phases. I am highly indebted to them for his help in solving my difficulties, which came across during this project stage on “Development of a mechanism for converting non-native CAD data while working with Heterogeneous CAD data”. I extend my sincere thanks to all teaching and non-teaching staff of department for their kind support and encouragement.

REFERENCES

[1] M. Mantyla, "A modeling system for top-down design of assembled products" in IBM Journal of Research and Development, Vol. 34, Issue 5, pp. 636-659, 2010.

[2] Y. Zeng, "Mathematical Foundation for Modeling Conceptual Design Sketches" in Transactions of the ASME, Vol. 4, pp. 150-160, 2004.

[3] A. Csabai, "Container spaces and functional features for top-down 3D layout design", in Computer-Aided Design, Volume 34, Issue 13, pp. 1011–1035, 2002.

[4] PTC Inc. internal library, “The Skeleton Modelling Approach”, Unpublished.

[5] Guk-Heon Choi, "Exchange of CAD Part Models Based on the Macro-Parametric Approach", in International Journal of CAD/CAM, Vol. 2, No. 1, pp. 13-21, 2002.

[6] Jinggao Li, "CAD Data Exchange Using the Macro-Parametrics Approach: An Error Report" in International Journal of CAD/CAM Vol. 10, No. 2, pp. 134-148, 2011.

[7] W. Li, W. Lu, J. Fuh, Y. Wong, Collaborative computer-aided design research and development status, Computer-Aided Design, Vol. 37, pp. 931–940, 2005.

[8] T. Mori, M. Cutkosky, "Agent-based collaborative design of parts in assembly" in Proceeding of ASME Design Engineering Technical Conference, 13–16, September, 1998.

[9] L. Chen, Z. Song, L. Feng, "Internet-enabled real-time collaborative assembly modeling via an e-assembly system status and promise" in Computer-Aided Design, Vol. 36, pp. 835–847, 2004.

[10] Y. Jin, W. Li, "Design concept generation: A hierarchical coevolutionary approach" in Journal of Mechanical Design, Vol. 129, pp. 1012–1021, 2007.

[11] Y. Zeng, A. Pardasani, J. Dickinson, Z. Li, H. Antunes, V. Gupta, D. Baulier, "Mathematical foundation for modeling conceptual design sketches" in Journal of Computing and Information Science in Engineering, Vol. 4, pp. 150–159, 2004.

[12] C. Tian, M. Masry, H. Lipson, "Physical sketching: Reconstruction and analysis of 3d objects from freehand sketches" Computer-Aided Design, Vol. 41, pp. 147–158, 2009.

[13] A. Csabai, I. Stroud, P. Xirouchakis, "Container spaces and functional features for top-down 3d layout design" in Computer-Aided Design, Vol. 34, pp. 1011–1035, 2002.

[14] N. Aleixos, M. Contero, et al., "Integrated modeling with top-down approach in subsidiary industries" in Computers in Industry, Vol. 53, pp. 97–116, 2004.

[15] Jinsang Hwang, "Representation and Propagation of Engineering Change Information in Collaborative Product Development using a Neutral Reference Model" in Korea Advance Institute of Science & Technology, July 14, 2010. cer.sagepub.com

[16] Ari Rappoport, "An Architecture for Universal CAD Data Exchange" in Proceedings of the eighth ACM symposium on Solid modeling and applications, pp. 266-269, 2003.

[17] In-Ho Song, "An XML-based Digital Mock-up System for Heterogeneous Assembly" in Journal of Mechanical Science and Technology, Vol. 21, pp. 1986-1993, 2007.

[18] L. Patil D. Dutta R. Sriram. Ontology-Based Exchange of Product Data Semantics. IEEE Transactions on Automation Science and Engineering, 2(3):213-225, 2005.

[19] PTC Inc. internal library, “The Skeleton Modelling Approach”, Unpublished.

[20] Mangesh Bhandarkar, “STEP-based feature extraction from STEP geometry for Agile Manufacturing” in Computers in Industry, Vol. 41, pp. 3-24, 2000.Jinggao Li, "CAD Data Exchange Using the Macro-Parametrics Approach: An Error Report" in International Journal of CAD/CAM Vol. 10, No. 2, pp. 134-148, 2011.

Jude A. Rodrigues, received the B.E. degree in Mechanical Engineering from Mumbai University, Maharashtra, India in 2009. He is currently pursuing his Master’s Degree in Computer Aided Design, Manufacture and Engineering (CADME) from ZCOER, Savitribai Phule Pune University, Maharashtra, India.