Improved Interoperability between MCAD and ECAD Design Tools

Robb McCord, Kelli Hosier, Doug Renwick, Mike Rindos
Northrop Grumman Mission Systems
Linthicum, Maryland

Abstract
During the design of printed wiring boards, our engineers exchange files between the MCAD and ECAD tools in order to synchronize boards and to validate the fulfillment of design requirements. Because numerous iterations of "design then check" involve the generation and then transfer of two types of data exchange format files, and because file management, and therefore configuration control, was mostly absent, this activity was identified as prime for process improvement. Therefore, a custom application was developed to add automation and configuration control to the interoperations between our MCAD and ECAD systems. For the MCAD capability, a vendor supplied programming language application was developed to create new menu items which allow users to skip most of the menu picks previously required to import and export files. In addition, the program supports the automatic identification and tagging of geometric elements such as board outline, keep-outs and coordinate systems with appropriate data exchange attribute types. The ECAD vendor's Application Programming Interface (API) was used to create new menu items in the ECAD tool that allow users to import and export files with pre-set standard options, instead of having to set dozens of options and layers manually. Utilization of our application allows our mechanical and electrical engineers to focus on design rather than the mechanics of generating, transferring and controlling files, thus trimming valuable time from the printed wiring board design process. This paper will include a process overview including background, use cases, development, and implementation process and cycle time savings.

Introduction
During the design of a printed wiring board, the ECAD and MCAD systems are usually operating as separate silos of automation that nevertheless need to exchange information as the process proceeds. The purpose of the exchange is to initially synchronize or to re-synchronize the board information between the 2 systems. Categories to synchronize include board geometry, mounting holes and keep-out and keep-in areas. Once synchronized on MCAD the user will check the board interface to the surrounding structure including chassis, heatsinks and mounting hardware. Also component placement will be checked for interferences to hardware, connectors and other components. Finally, routing checking will be performed to check between copper routes on outer layers, hardware and surface traces. The initial synchronization from MCAD to ECAD allows the ECAD user to create the board geometry as it was created on the MCAD system.

The two primary data formats used for MCAD/ECAD interoperability are Data eXchange Format (DXF) and Intermediate Format for Data (IDF). A depiction of the data represented in these formats is shown in Figure 1.

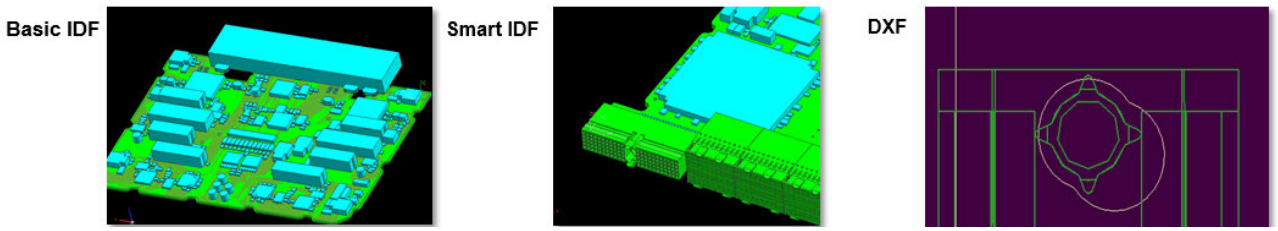


Figure 1 – IDF and DXF Comparison

For our application generally DXF is used in the 2D wireframe mode and is used primarily in the overlay mode to check geometry between the mechanical and electrical models. DXF also supports extensive layering capabilities so the often robust layering conventions used in the design tools do convey back and forth between the two systems. DXF objects are graphical in nature containing information about all board objects including copper objects but contains no associated intelligence.

For this application the IDF file contains a mix of 2D and 3D objects and also includes intelligence that can be interpreted by the CAD tools. The IDF file will contain board and component information but copper objects are not currently supported by the CAD tool translators. IDF can be used in two major modes of operation: Basic or Smart. The Basic mode conveys

basic geometric shapes between the systems including board shape and thickness with holes, cutouts and millings. Also included are 3D prismatic shapes representing board components with the extrusion heights based on the maximum library part dimensions. The Basic IDF mode typically includes two separate files to complete an IDF data set – the board file and the library file containing component locations, dimensions and extrusion heights. In the Smart IDF mode there is no separate library file, rather the actual component files are pulled in from the CAD system's library during the IDF import. This method requires synchronization of the ECAD and MCAD libraries in geometric dimensions, origin, orientation and name and as such represents a very high level of integration rarely achieved due to the volume of legacy library parts that would have to be reworked or created anew.

Finally, the potential in improving MCAD to ECAD interoperabilitywas highlighted by Dale Parker when he included better data exchange in his "10 Rules for Improving PCB Design". He says "Data exchange with mechanical tools and electromechanical designs absolutely needs to be improved. The smaller size and increased electronic and mechanical complexity of IoT and wearable products means that packaging and electronic assemblies are becoming more tightly integrated than ever before. Teams working on the electronic and mechanical aspects of the design need to continuously exchange design information to make sure that they are not wasting time on a direction that will be physically incompatible."[1]

Background and Overview of Old Process

During the design, in order to achieve the synchronization described above in the introduction, many files are exchanged in an iterative manner by the designers. In the old process the files were manually created on the MCAD and ECAD systems and then transferred in an ad-hoc manner to the users on the other system in a variety of ways including email and shared drives. Multiple copies of what looked like the same files existed at the same time and there was no version or configuration control of any kind and costly mistakes and rework occurred. The entire process was highly dependent on users keeping track of what file was sent and where it was located. The old process is summarized humorously in Figure 2 below.

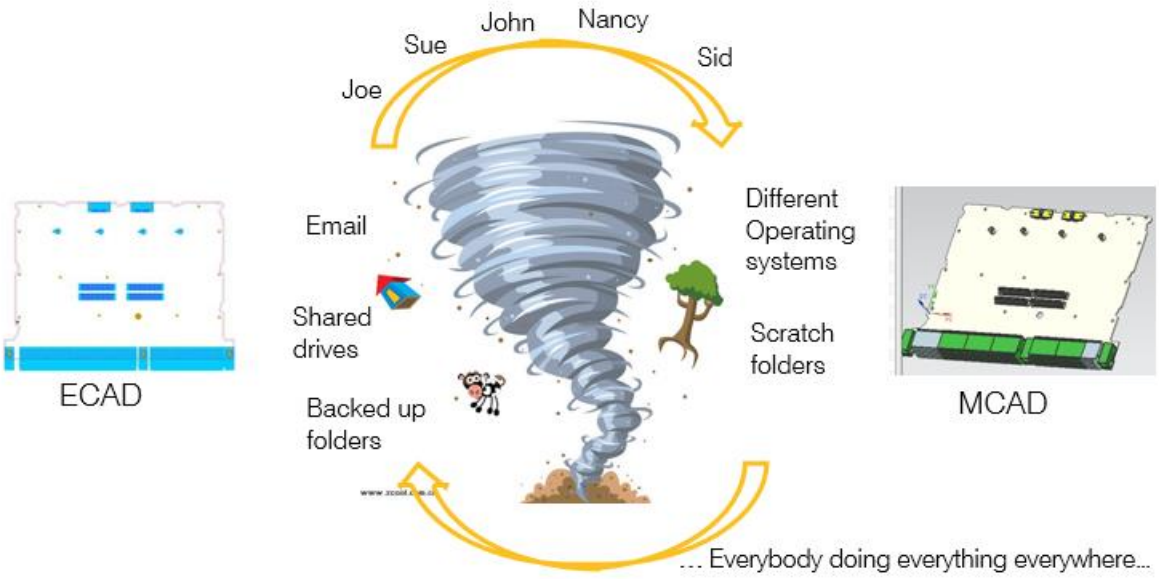


Figure 2 – The Old Process

A team was formed to map out the process in place for file exchanges and to look for automation possibilities to put the process under better control. The team selected the name One Board to denote the new process. The author has used One Board in this document when referring to the new process. The almost immediate findings were that the biggest payback would be in eliminating manual interaction with files and storing files in a central repository under configuration control.

The Development of an Improved Process

A prerequisite for improving the process was identified as shortcomings in the IDF translators of both the MCAD and ECAD systems. For example, layer names containing spaces or mixed case letters coming from the MCAD system caused import problems on the ECAD system. Also internal routing vias included in the DXF files exported from the ECAD system caused very long import processing times on the MCAD system. In addition, different IDF version (V2 or V3) and file extensions between the systems cause additional interoperability problems.

The solution to these IDF issueswas to develop a custom post processor that would properly flavor the IDF being transferred to be more compatible with the receiving system. During processing any layer names with spaces or mixed case letters are automatically corrected. Also internal vias are optionally removed (ECAD to MCAD only) and the IDF board file and library files are renamed to fit the file extension desired by the receiving system. Figure 3 shows the custom post processor developed and how it improves MCAD <> ECAD transfers through flavoring. The post processor can be accessed interactively by a user with a Graphical User Interface or via a command line call for use by programs.

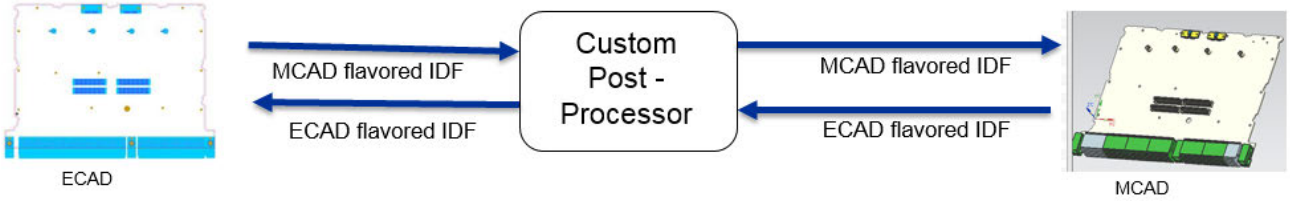


Figure 3 – Custom Post Processor for Flavoring

With a solution to basic IDF compatibilities in place the team moved on to developing a flowchart of the current process which turned out to be a non-trivial task. After many meetings, the users of the MCAD and ECAD systems reached a consensusand a flow chart of the current process was created. Key elements of the flow chart are shown in Figure 4.

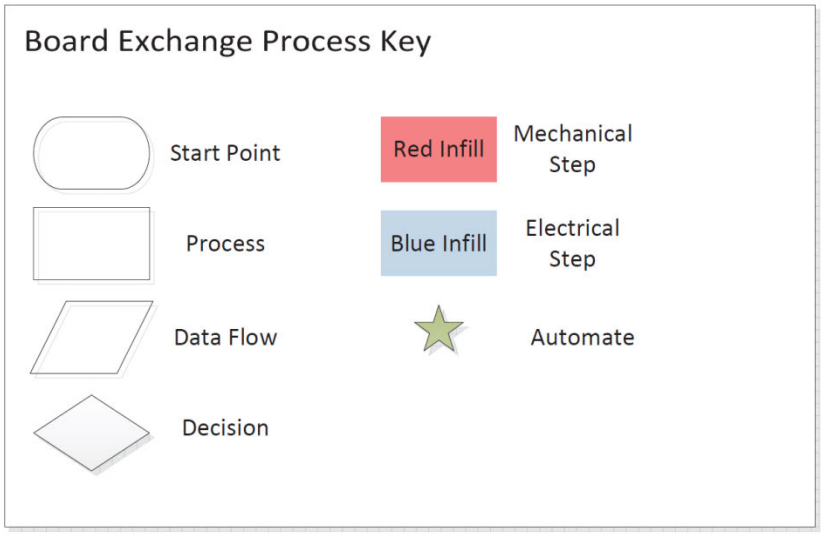


Figure 4 – Flowchart Elements

Figure 5 maps out the flowchart, but it is not necessary to read all the captions in the boxes at this point. The key takeaways are that the MCAD board design and ECAD board design, component placement, board routing and component adjustments are supported by 2 iterative file transfer loops.

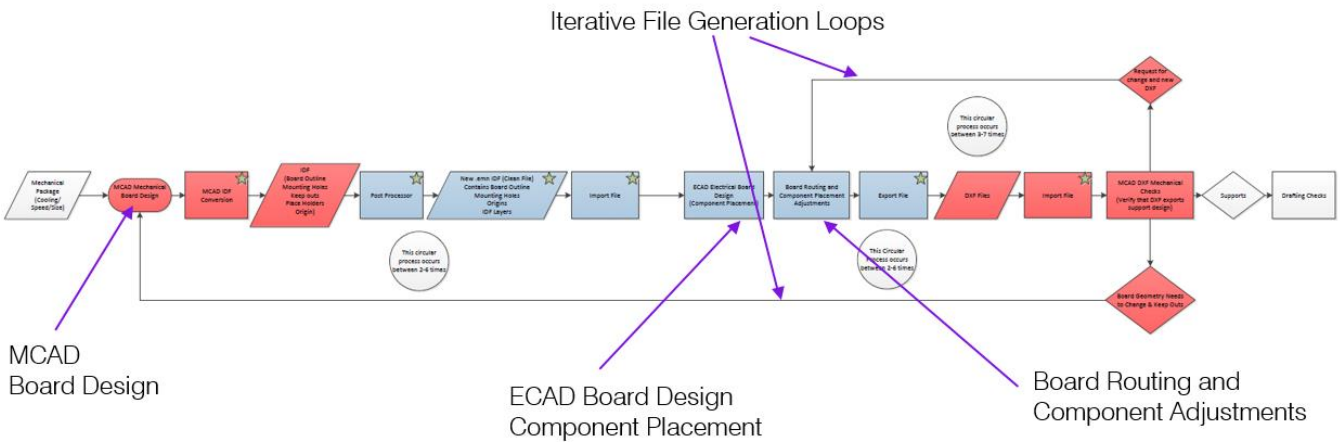


Figure 5 – Old Process Flow

The DXF iterative file transfer loop is shown in Figure 6. The ECAD user completes board routing and component adjustments and exports a new DXF file to be used for checking in the mechanical CAD system. Board and component geometries, copper layout and origins of components and holes from the ECAD system are compared to the mechanical layout and design requirements. If changes are required this is communicated back to the ECAD user and a new DXF for checking is generated. This process can occur 5 to 7 times during the board design.

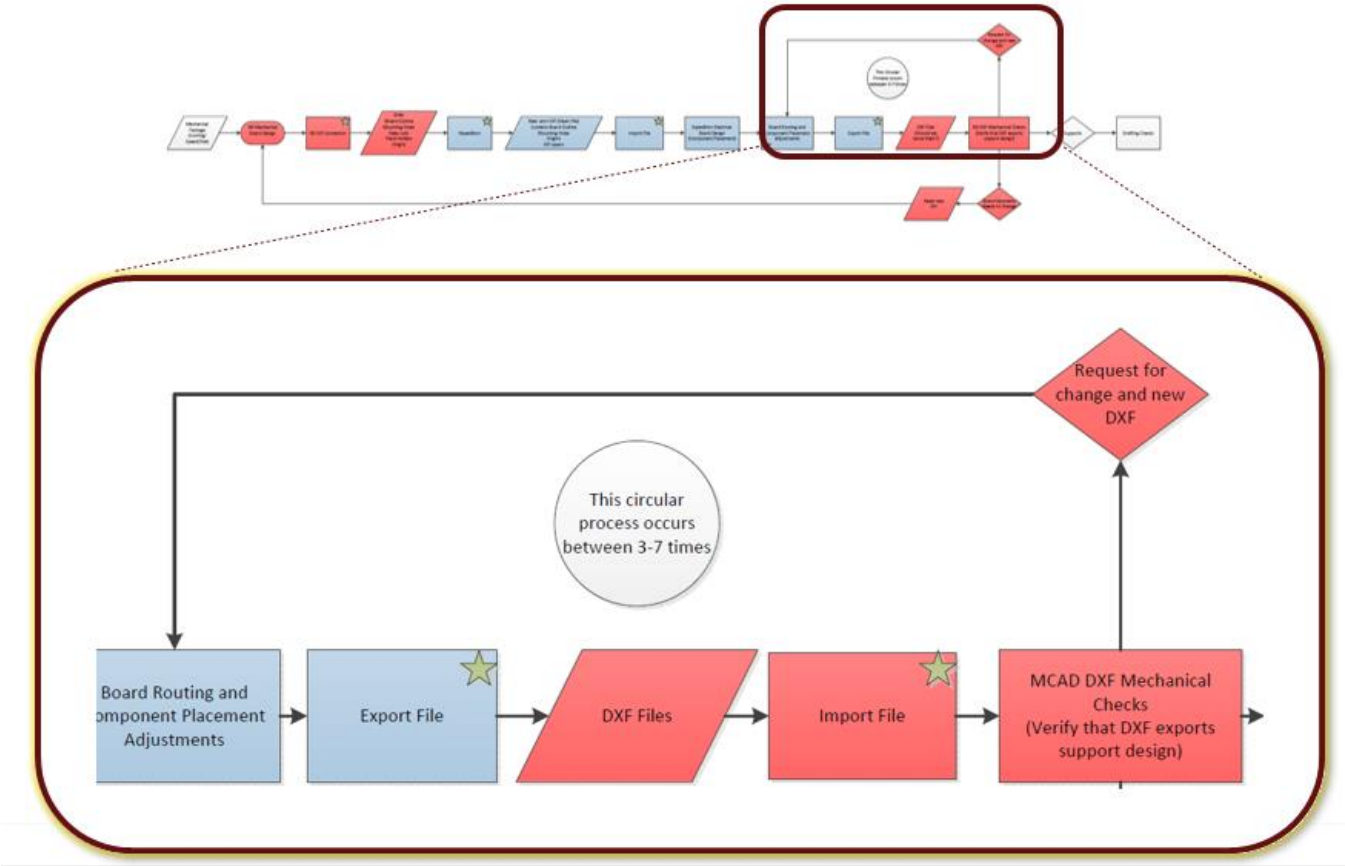


Figure 6 – Detail of the DXF Iterative Loop

For a summary of the IDF iterative file transfer loop it is best to refer to Figure 5 where the long line across the bottom represents this loop. When board geometry or keep-outs need to change a request is made for a new IDF from the MCAD tool and the new IDF is produced and processed like the initial IDF file. This iteration takes place about 3 to 7 times for each board design.

The first cut solution to solving the problems of manually generating and handling files was developed by the OneBoard team and approved by the user community. First a number of programming routines and scripts would be created in the CAD tools using API programming to perform most of the tasks involved in importing and exporting files. The programs would take care of as much of the user selection steps as possible. For MCAD IDF export the program would automatically assign board solid and all keep-outs upon running. Taking into account the advantages and disadvantages of each format, OneBoardhas standardized on IDF for ECAD to MCAD transfers and on DXF for MCAD to ECAD. Finally, a commercial file storage solution was selected as a common repository for data.

Overview of New Process and Implementation

The One Board automation was developed and tested over a several month long period and then pilot tested with 3 different sets of users. At the heart of the application is the One Board file repository where files are automatically placed and versioned by the automations built into the ECAD and MCAD tools. Files can be accessed in the repository via a web browser, drive mappings orUniversal Naming Convention(UNC) style pathnames (example: \\servername\folder1\folder2). A snapshot of the One Board file repository is shown in Figure 7.

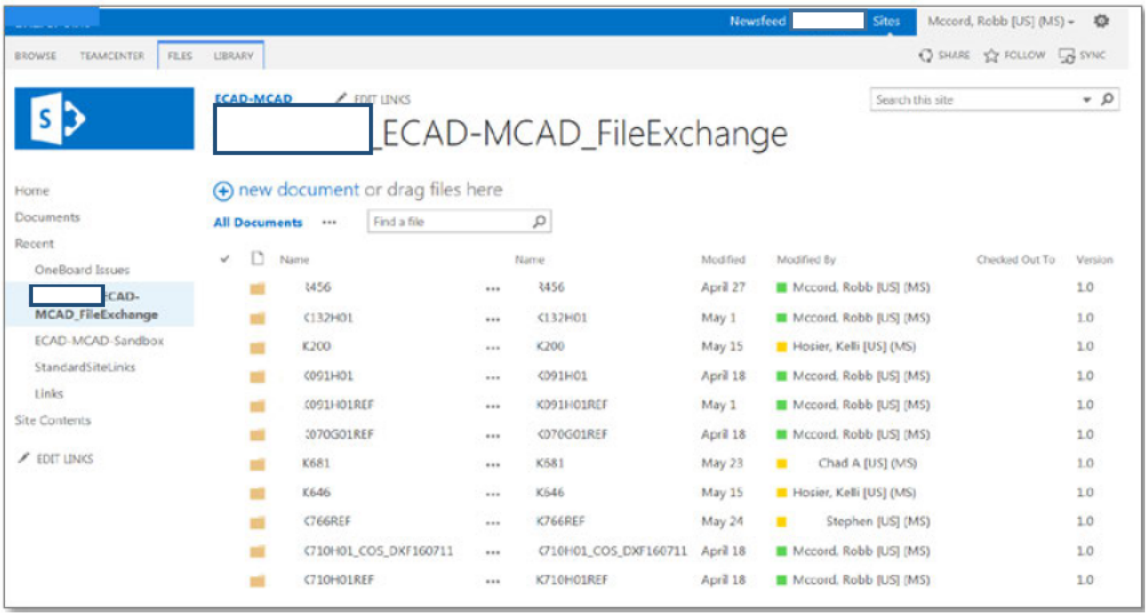


Figure 7 – Top Level File Repository

The MCAD automation for generating the first IDF file of the process was created as a combination of in-house developed API programming along with a vendor supplier routine for outputting the IDF file. A number of functions related to gathering the required data from the work part, removing any old attributes and writing out the file are performed. The program automatically assigns the IDF board outline tag to the main solid body of the work part and "other outline" tags to all sheet solids. The default name for the other outline sheets is always place keep-out. Also the board coordinate system is derived by the program from the current work part coordinate system. After initial tag assignment a user interface is displayed allowing users to change the other outline names if desired. After closing the user interface the IDF is written to the One Board file storage system where files are version controlled and cannot be deleted or overwritten by the users. The IDF file will contain the board outline, cutouts, keep-outs, other outlines (default name of place keep-out) and board mounting and component placement guide holes. Figure 8 shows an overview of the MCAD automation with basic functions performed by each Dynamic Link Libraries (DLL) created defined.

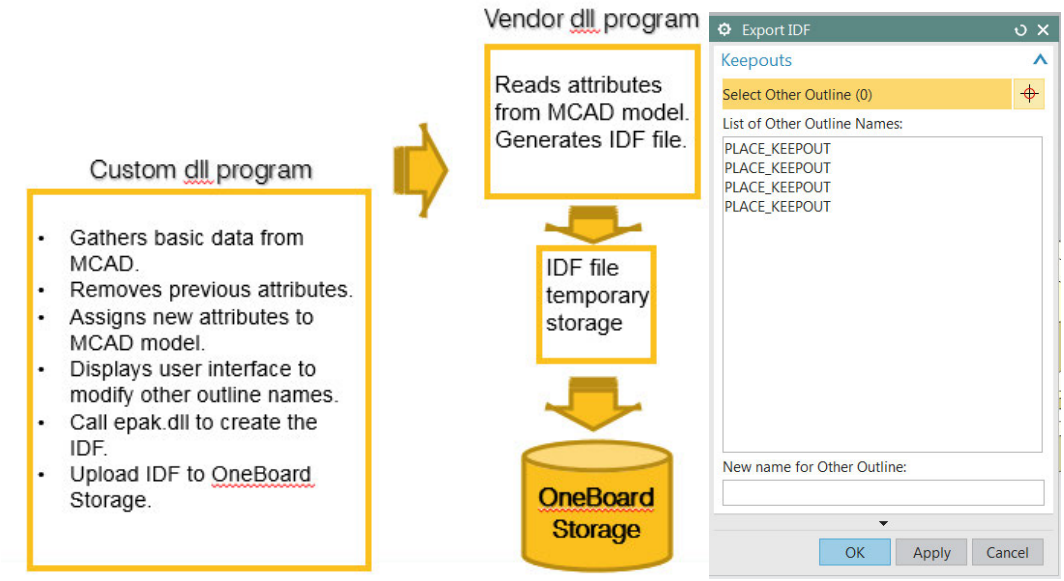


Figure 8 – MCAD IDF Export Automation

The ECAD automation for importing IDF first looks to the One Board file storage and will pull the most current version of IDF for the part number of the board being worked. Next the custom post processor is run in background to flavor the file per the guidelines discussed in the "**The Development of an Improved Process"** section. The CAD tool vendor supplied IDF import command is then used toimport the file to the ECAD system. Issues with incoming IDF files that used to cause import problems such as location of the board origin, how cutouts and slots are defined and types of characters in the layer names are automatically handled by the custom program and the post processor.

To communicate revised component locations, moved holes and changed board outline back to the MCAD user, the ECAD user utilizes the DXF export function that was developed. This automation generates a DXF file containing all part and copper objects from the top and bottom layers of the board. This file will contain all of the objects required to geometrically describe the board such as board outline, origin and holes in a simple 2D wireframe format.

The MCAD automation for importing the DXF file was created to pull the most current revision and version file from the One Board storage system automatically and import it in the work part. The guess work of which file is the most current and the chances for error are eliminated. The files can be very large and include only "dumb" graphics with no attributes included. However they meet the requirements for checking the state of mechanical interfacing and component locations between the MCAD and ECAD systems. The DXF files have the advantage of depicting the exact geometry of the components whereas in the IDF data components are represented by the placement outlines in ECAD library which are typically larger than the real geometry.

The final step in developing One Board was the selection of a commercial file management system. Commercial file management "System A" was already widely used within our company for controlling in process and released MCAD files but had not yet been deployed to our ECAD users. One problem in using it for One Board was in licensing. System A is licensed "per user" and not enough licenses existed (or budget to buy more) for all the engineers that would need to use One Board. Also, in comparing ease of use, System A required many more steps in navigating and finding files than did file management "System B". In addition to being easier to use System B was already licensed for every user of the company so it was selected.

Implementation

One Board was collaboratively created over a 3 month period by a team of 8 CAD developers and expert MCAD and ECAD users. The most challenging task was the agreement on and creation of the as-is process flow chart shown in Figure 5. After agreement the applications required were developed using custom code in the API languages of the MCAD and ECAD systems. After completion, 3 separate pilot programs of small groups of users were conducted with resulting constructive feedback being channeled back into improvements to the system. The last step was the successful rollout to approximately 800 MCAD and 500 ECAD users. Metrics in usage and savings are still in preparation. Figure 9 shows a high level representation of the One Board system.

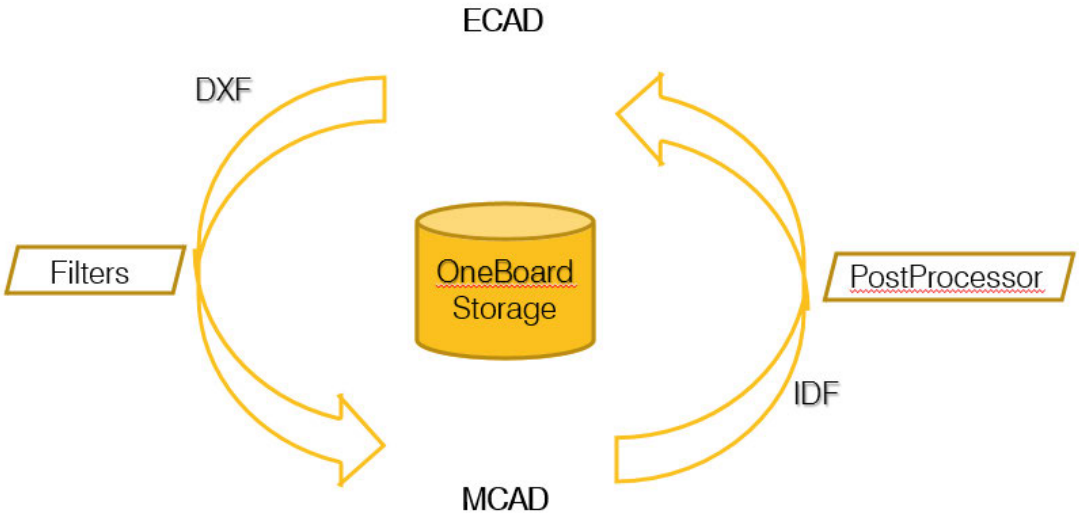


Figure 9 – One Board: Improved MCAD/ECAD Collaboration

Conclusions and Future Goal

The implementation of the One Board project has enabled better collaboration between our MCAD and ECAD users during the design of printed wiring boards. It allows our mechanical and electrical engineers to focus on design rather than the mechanics of generating, transferring and controlling IDF and DXF files, thus trimming valuable time from the printed wiring board design process. The "One Button" approach to imports and exports where many levels of menu picks are eliminated greatly reduces human errors such as output files having the wrong version, file extension or units and eliminates manual tracking of revision and location of files. The automatic post processing and flavoring of files overcomes the deficiencies of the out of the box import and export translators of the MCAD and ECAD systems.

The next steps for the system are focused on incremental continuous improvement. One current project is to improve the DXF filtering from ECAD to MCAD to reduce the number crosshatch objects translated which are not required for the checking process thereby substantially reducing overall file sizes. Another area of potential improvement is the commercial file management system being used. Current plans are to migrate the ECAD users and the data into the System A previously discussed in the Overview section. Once that happens it is likely there will be synergies that can be exploited to improve the system so re-hosting One Board from System B to System A will be studied. Finally, we will continue to carefully examine any new technology, emerging data exchange format such as IDX or future vendor software offerings for opportunities to further refine the collaborative capabilities of the One Board system.

References

1. Parker, Dale (2016), Brother Can You Par-a-digm? , IPC Apex Expo Conference, Las Vegas, Nevada.

Improved Interoperability between MCAD and ECAD Design Tools

Robb McCord, Mike Rindos

Northrop Grumman Mission Systems

Agenda

- Overview of the project to better integrate the ECAD and MCAD systems nicknamed Oneboard
- Background
- Documentation of current process
- Solution Overview
- Status
- Future Plans

Overview

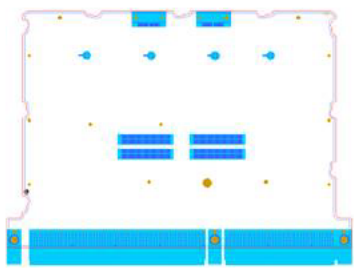
- Board data files are exchanged between MCAD and ECAD for many reasons
  - Synchronizing MCAD and ECAD representations of board: origins, mounting holes, outlines, keepout areas, etc.
  - Board checking: interface with surrounding structure, chassis, heatsinks, mounting hardware
  - Placement checking: interference between different parts, between parts and mechanical hardware, and between component/connector locations
  - Routing checking: interference between copper routes on outer layers and mechanical hardware, surface traces, shorting, thermal interfaces and RF/digital ground planes.

Problem and Proposed Solution

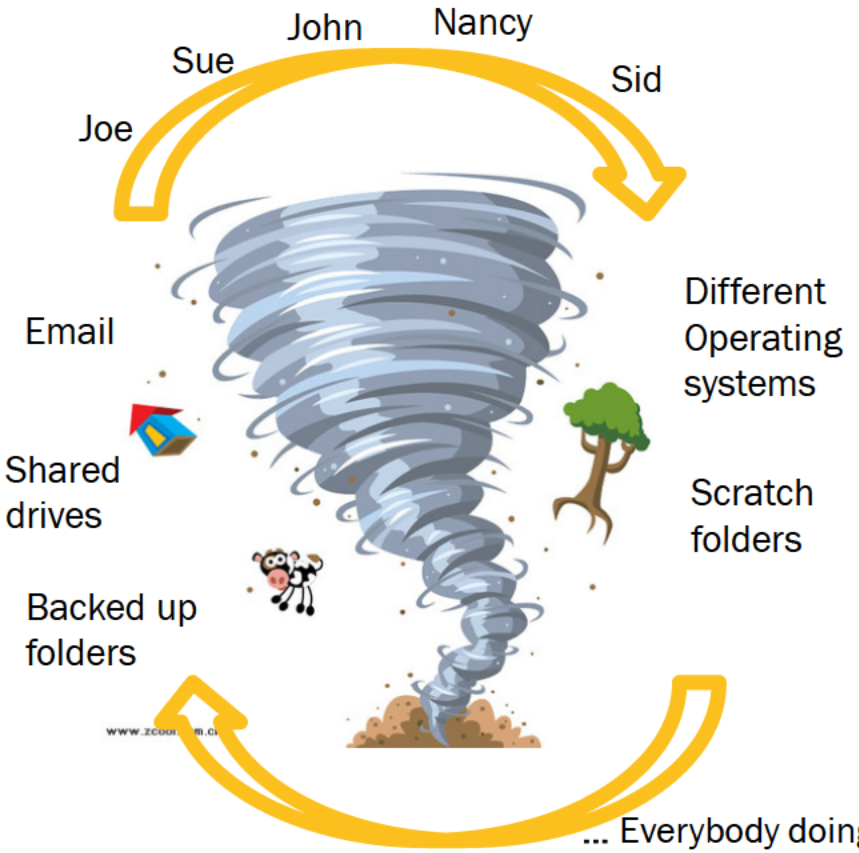
- This exchange has had problems
  - Files are exchanged through email
  - Files are manually renamed between the MCAD and ECAD systems
  - Multiple copies of files exist
  - Hard to tell if you have the latest version of the file
  - Intermediate Format for Data (IDF) files generated by the MCAD and ECAD systems have issues which cause the receiving system problems
- The Oneboard team was formed to address these issues
  - *Explore and map the current processes and look for automation possibilities focused on:*
    - Eliminating manual interaction with files
    - Storing files in a central repository
    - Maintaining configuration control on the files

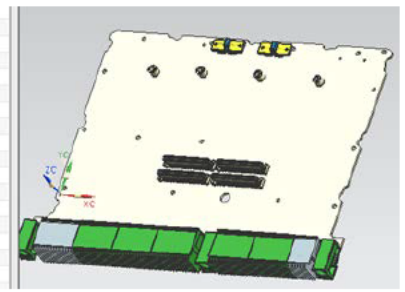
Past Process

- No configuration control; files not controlled….



ECAD





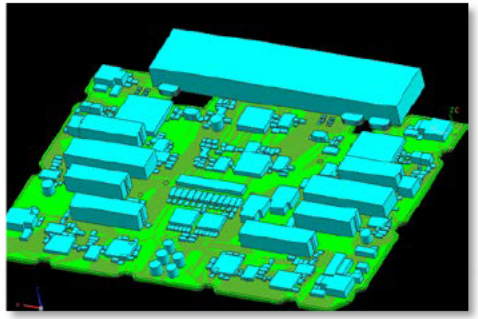
MCAD

… Everybody doing everything everywhere…

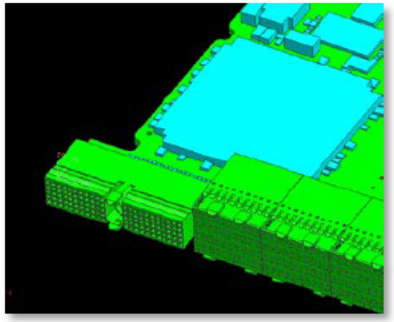
…Highly dependent on users keeping track of exchanges

Board Data Files Comparison

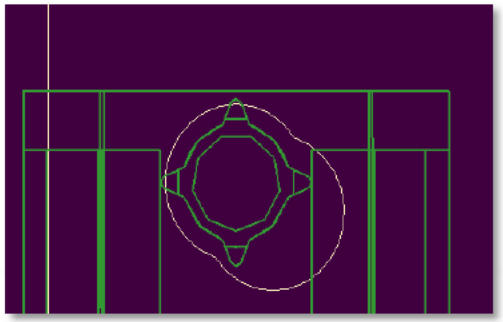
- IDF Files
  - Intermediate Format for Data– can be 3D
  - Specification for exchanging data between Electrical and Mechanical CAD tools
  - Objects contain intelligence that can be interpreted by the CAD tool
  - Contains board and part information, but no copper information
  - Optimal use when MCAD and ECAD libraries are synchronized (Smart IDF)
- DXF Files (Data Exchange Format)
  - 2D in our usage
  - Objects are graphical with no associated intelligence
  - Contains information about all board objects, including copper objects



Basic IDF



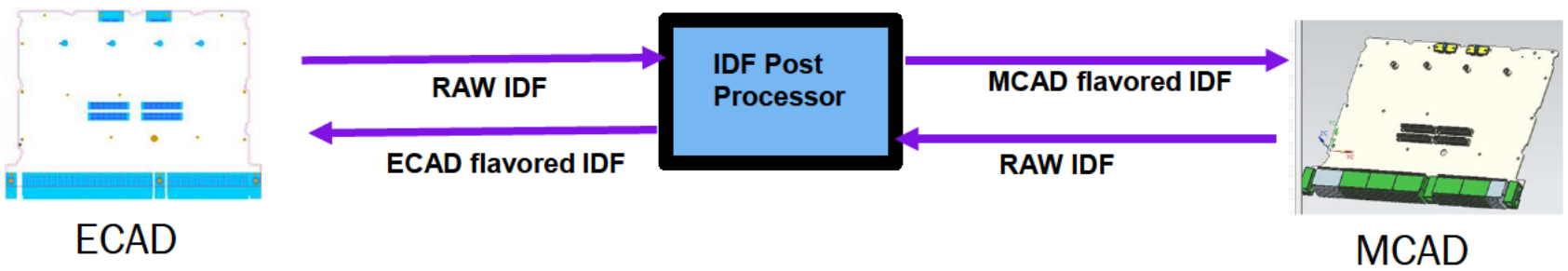
Smart IDF



DXF

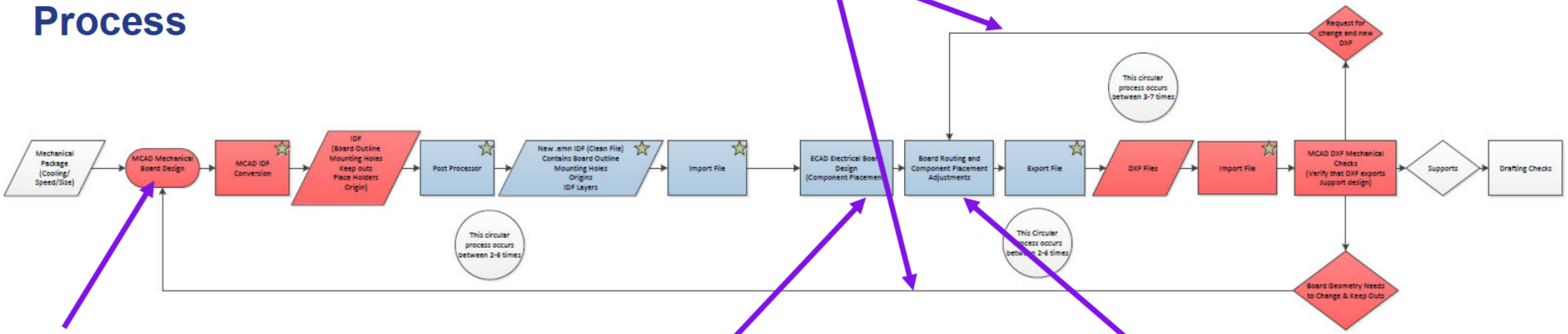
IDF Interoperation Considerations – PostProcessor Creation

- Layer names with spaces and MiXed case cause ECAD problems
- Internal routing vias cause huge files which cause MCAD problems
- The 2 systems use different file extensions for the IDF board and the library files and refuse to recognize any other extensions
- Others: duplicate board outlines, wrong IDF version, …
- Solution: custom developed post processor to process IDF into a format to better import into the receiving system



Mapping Former Process

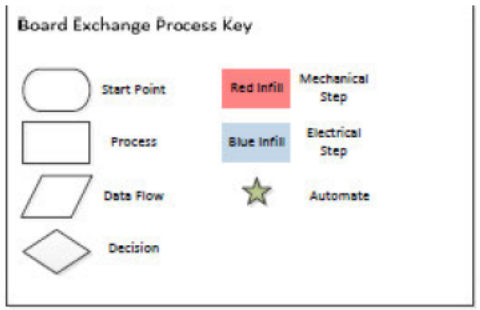
Iterative File Generation Loops



MCAD
Board Design

ECAD Board Design
Component Placement

Board Routing and
Component Adjustments

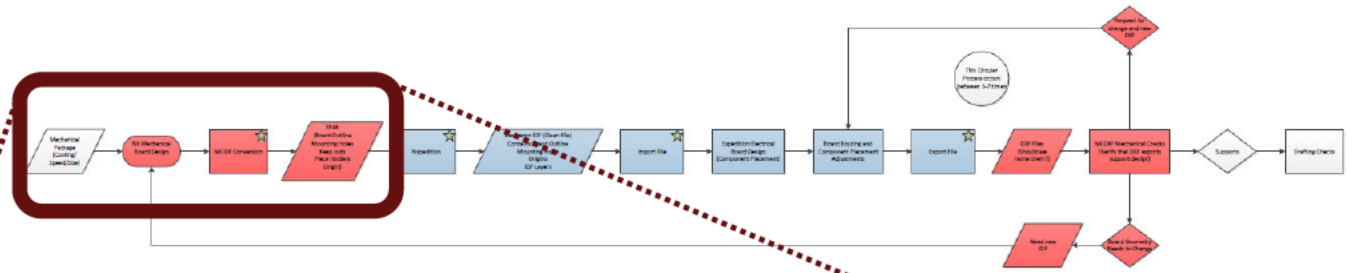


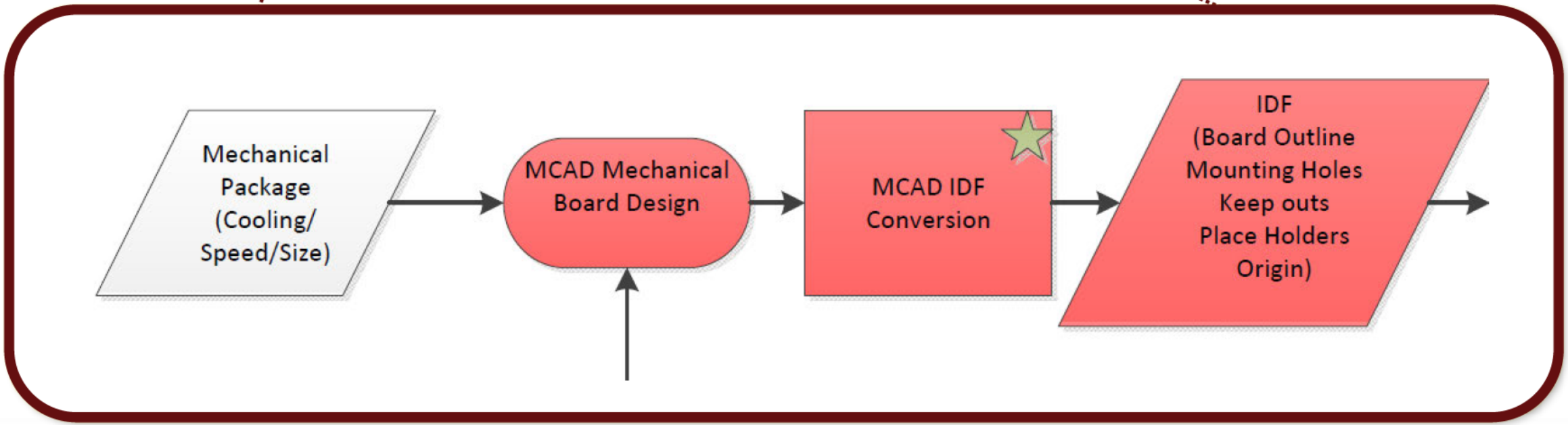
… Seems simple enough… Not… building consensus for this chart may have been the hardest step

Former Process Details

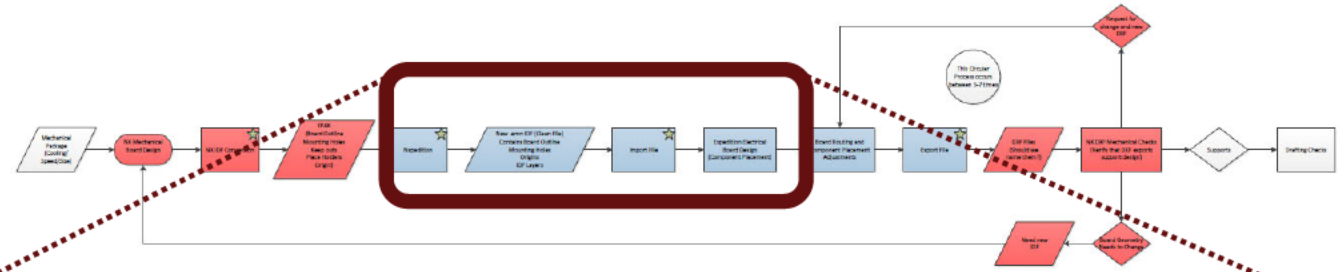
- The next 4 slides show the details of the former manual interfacing between ECAD and MCAD
- Gold Stars highlight the areas for automation identified by the team

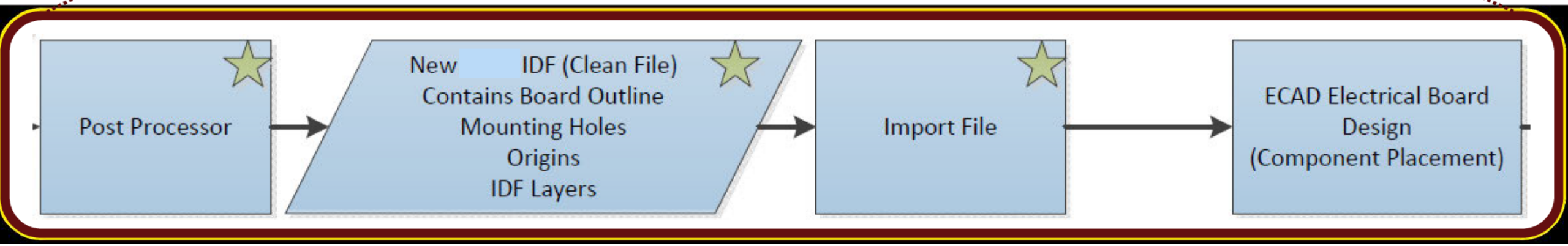
OneBoard Process – IDF Export



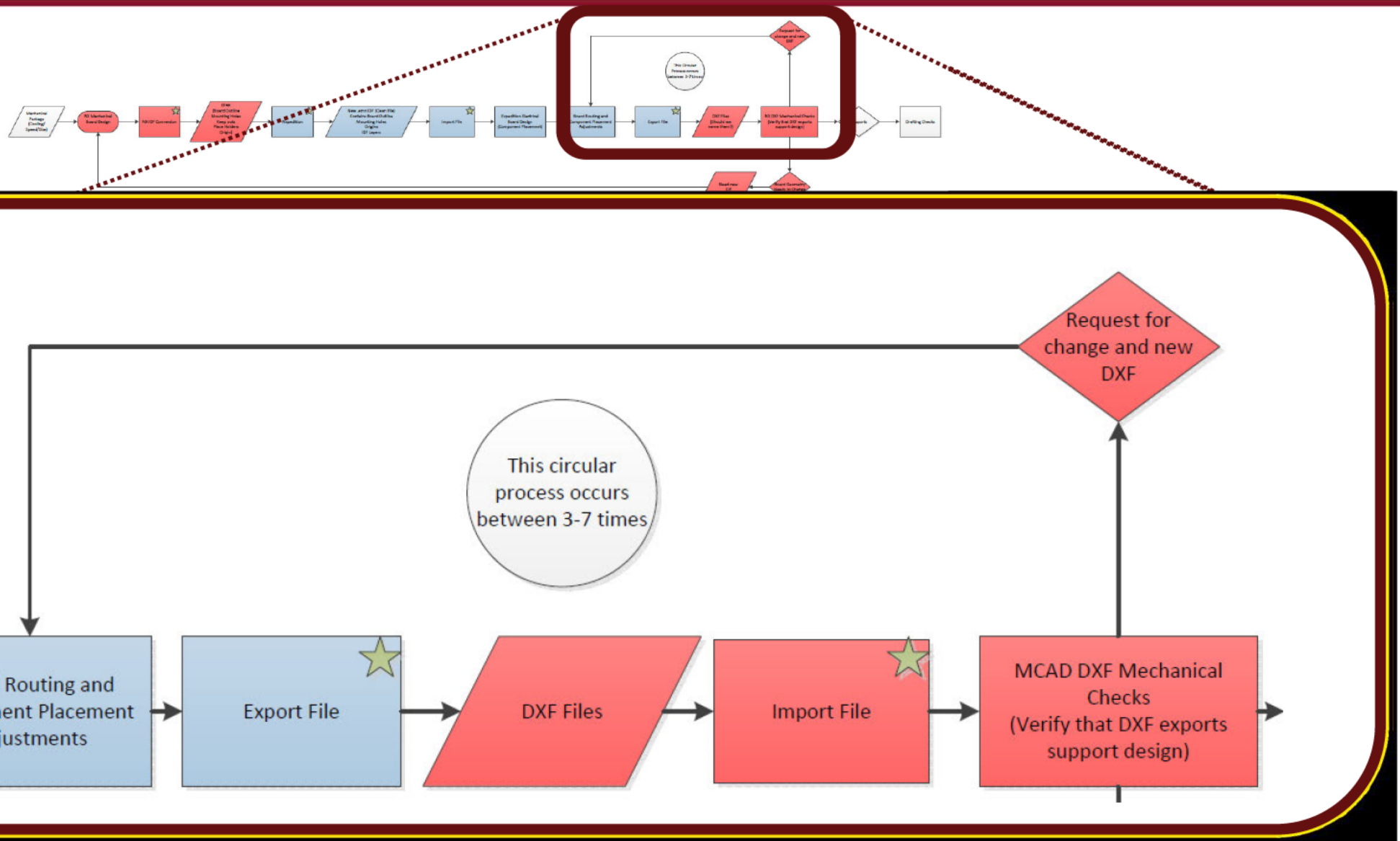


OneBoard Process – IDF Import

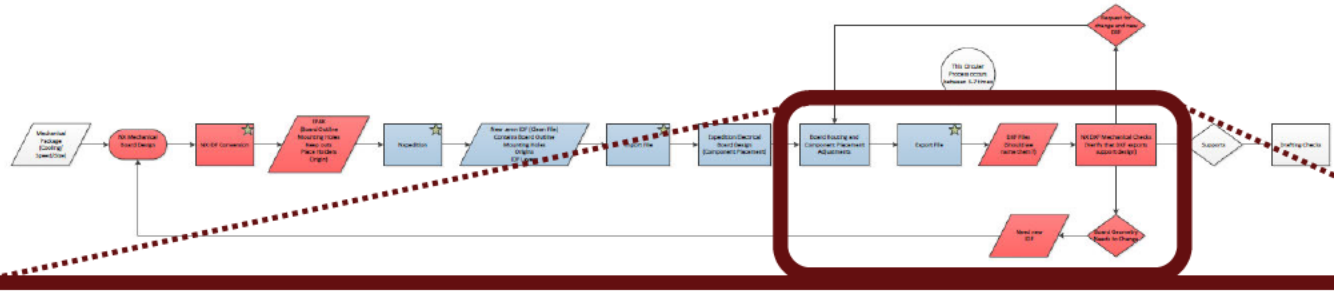


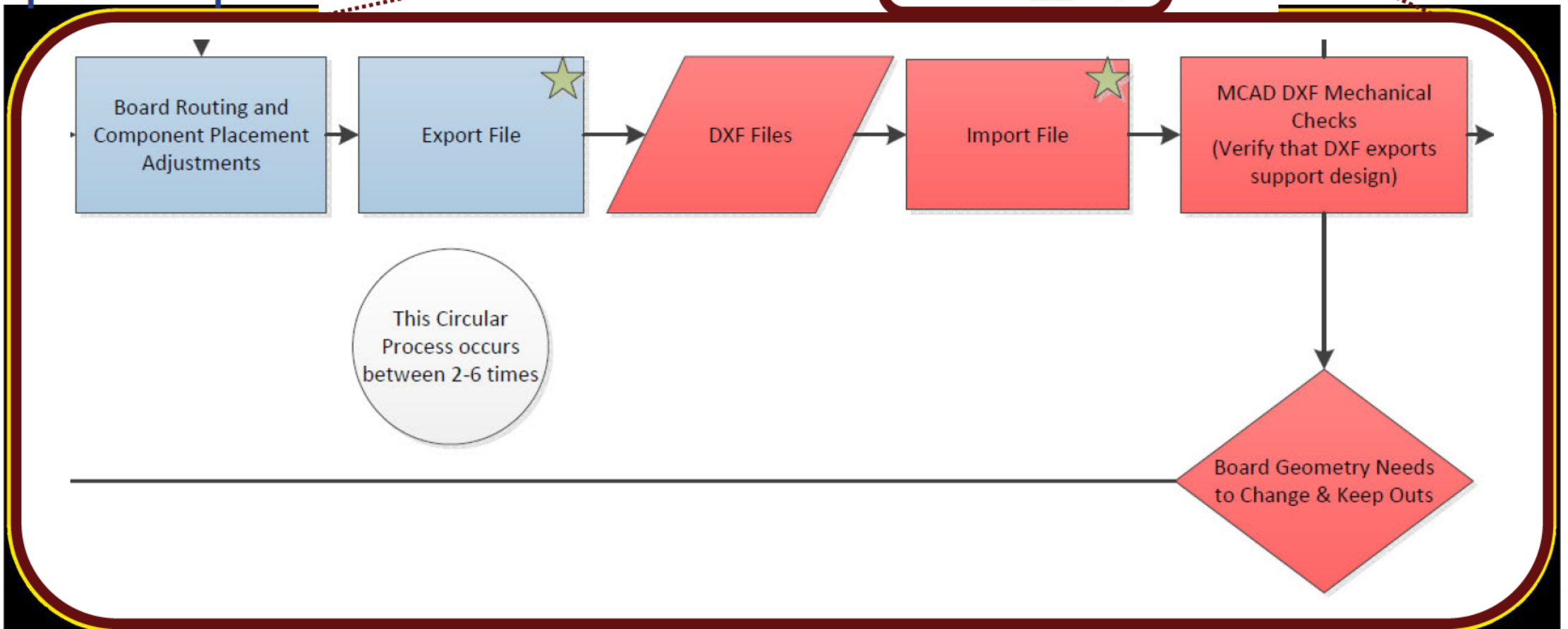


OneBoard Process – DXF Export/ Import



Oneboard Process – DXF Export/Import

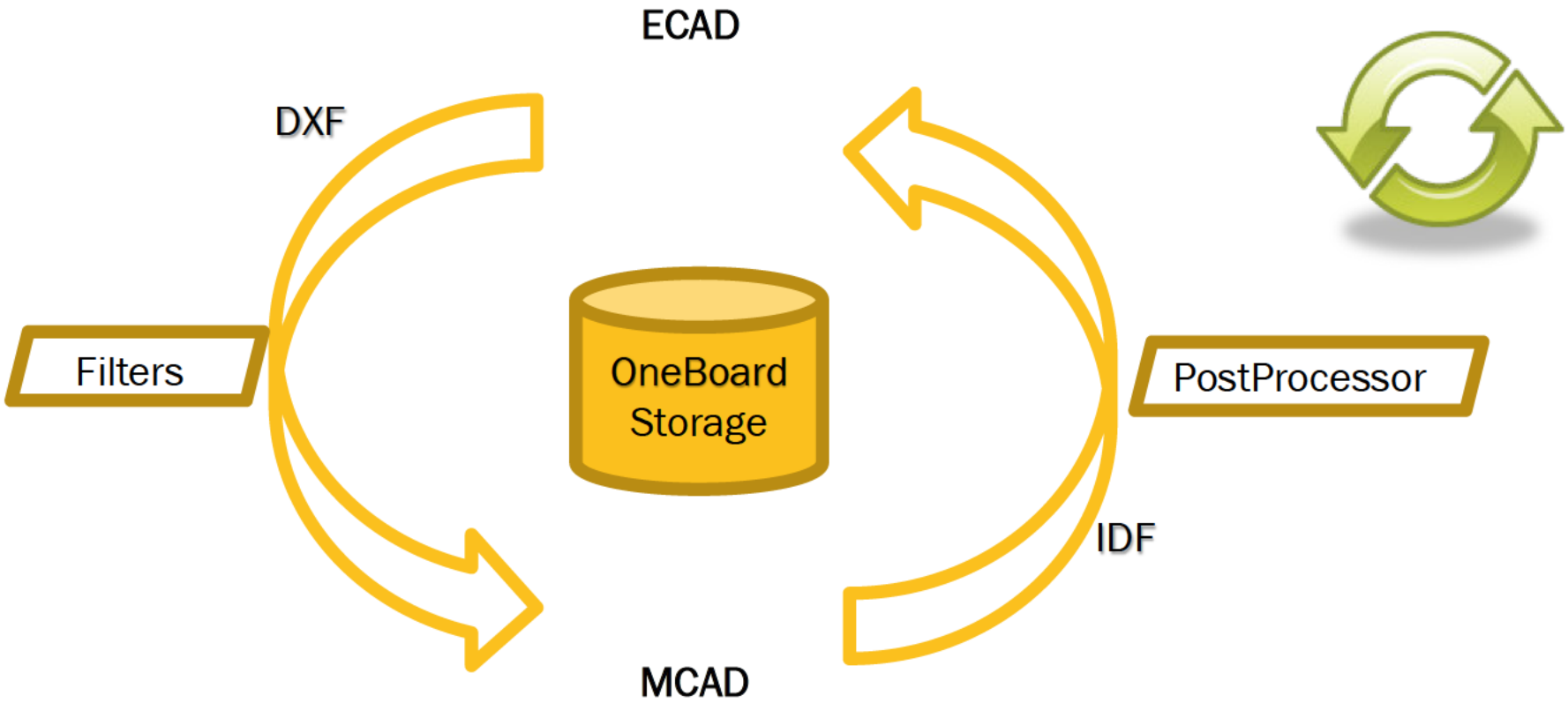




Oneboard Solution Overview

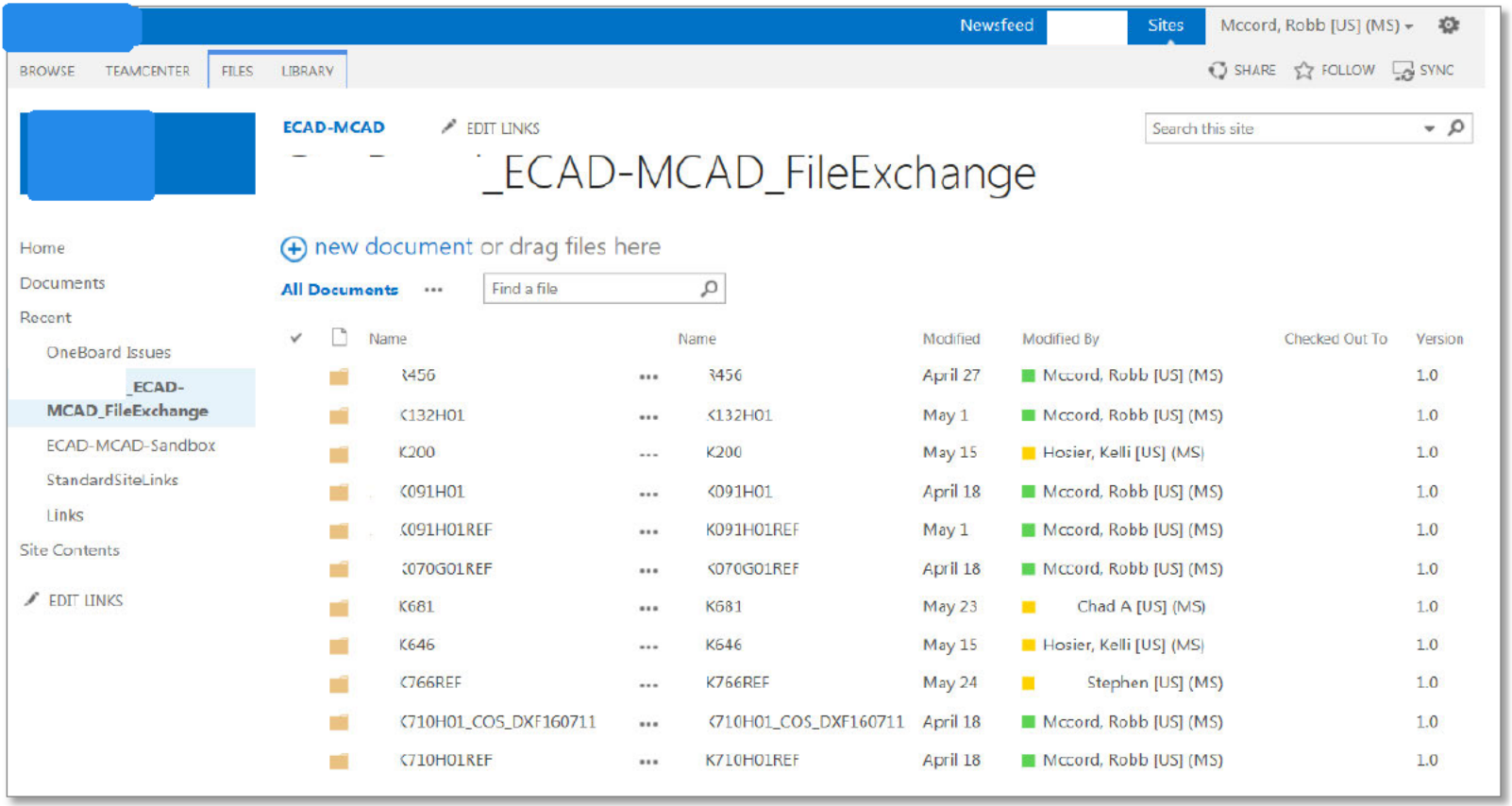
- Programs created to run within the ECAD and MCAD tools to automate the import/export functions during design interoperations
- IDF selected for MCAD to ECAD initialization and subsequent transfers
- DXF selected for ECAD to MCAD
- New menu picks with predictive behavior – with most common selections already populated or derived from the ECAD or MCAD models
- Many levels of menu selections of the prior process eliminated
- Configuration controlled file storage – permanent record of who, what and when

Oneboard Solution Overview



…Improved MCAD/ECAD Collaboration

File Repository

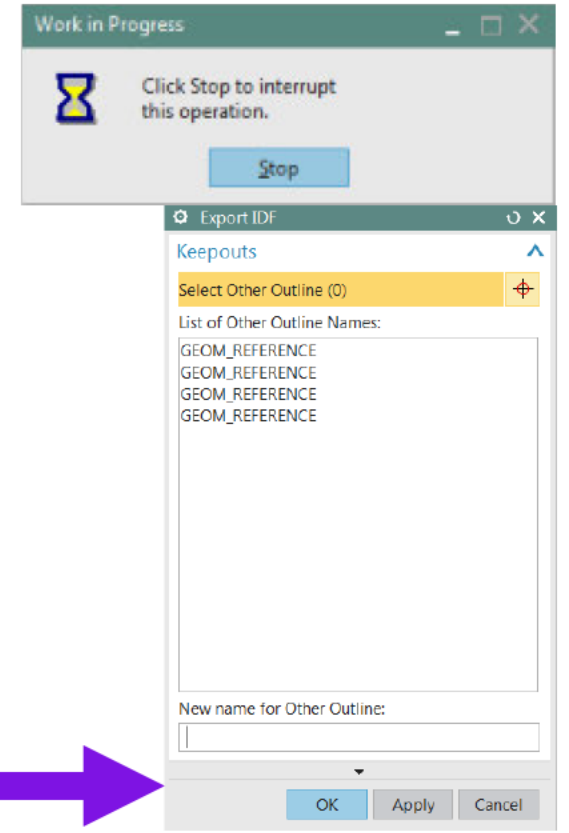


- ECAD/MCAD Collaboration site:
  - Upload or download via automations or manually
  - Access via online browser, drive mapping, Uniform Naming Convention (UNC) path

  Automatic naming and versioning by the custom programs
  - Uses commercial file management product

Customized IDF Export From MCAD

- Translators tab > Export IDF
  - Automatically detects board solid and keepout sheet solids
  - Derives board coordinates from current Work Coordinate System
  - Assigns Design Authority ownership and file name from MCAD part name
  - Generates IDF file of assembly
  - Writes the file to the OneBoard Storage system



IDF Data Exported

- Board Outline
- Cutouts
- Keepouts
- Other Outlines
  - Generic shapes put on user-defined layer IDF_<layer_name>
  - Can be converted to intelligent objects (keepouts) in the ECAD system
- Drilled Holes
  - Board mounting holes
  - Component placement guide holes

Customized MCAD Export Function

Custom dll program

- Gathers basic data from MCAD.
- Removes previous attributes.
- Assigns new attributes to MCAD model.
- Displays user interface to modify other outline names.
- Call epak.dll to create the IDF.
- Upload IDF to OneBoard Storage.

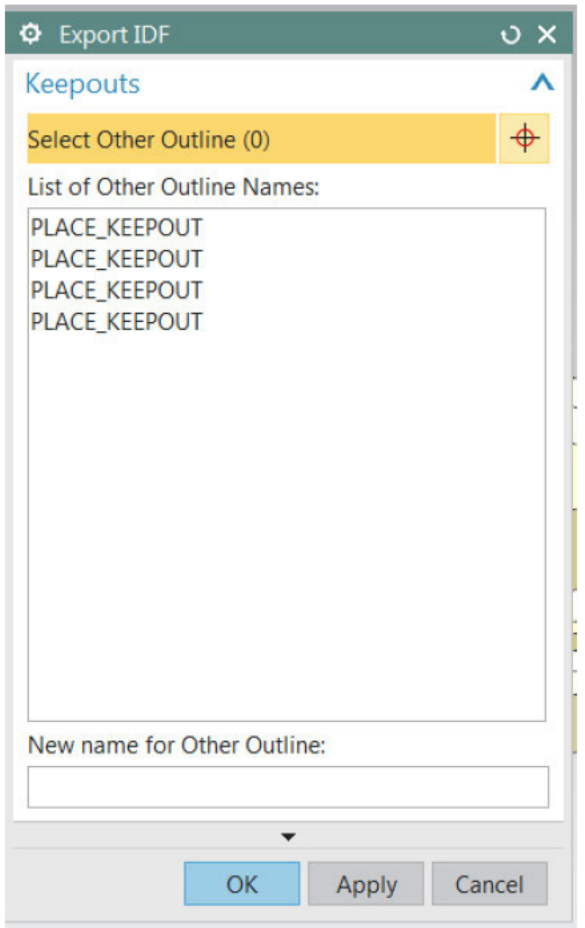
Vendor dll program

Reads attributes from MCAD model. Generates IDF file.

IDF file temporary storage

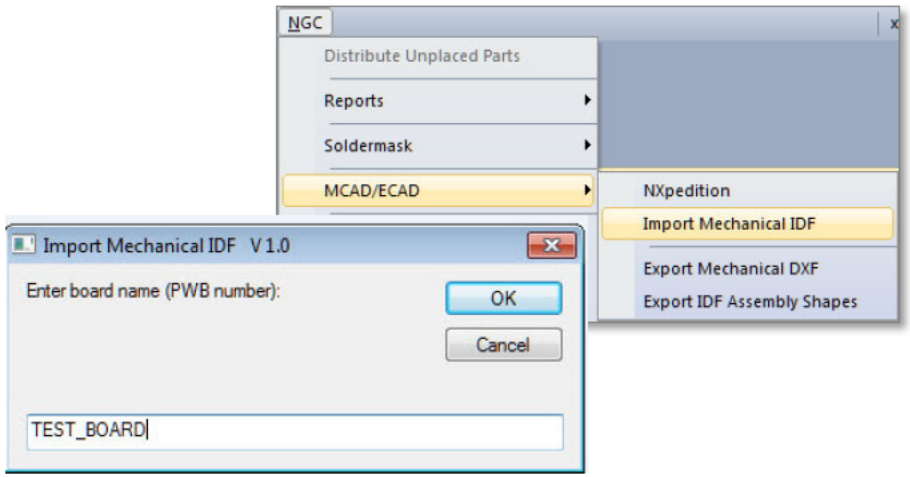
OneBoard Storage

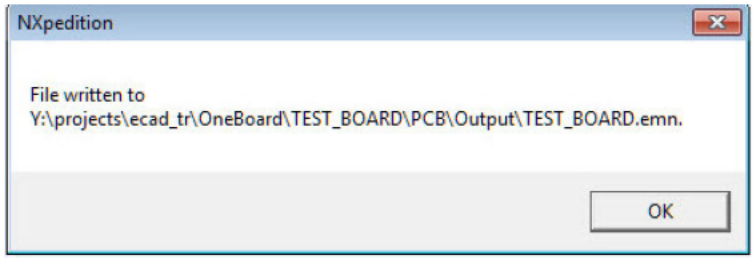
MCAD UI for Export

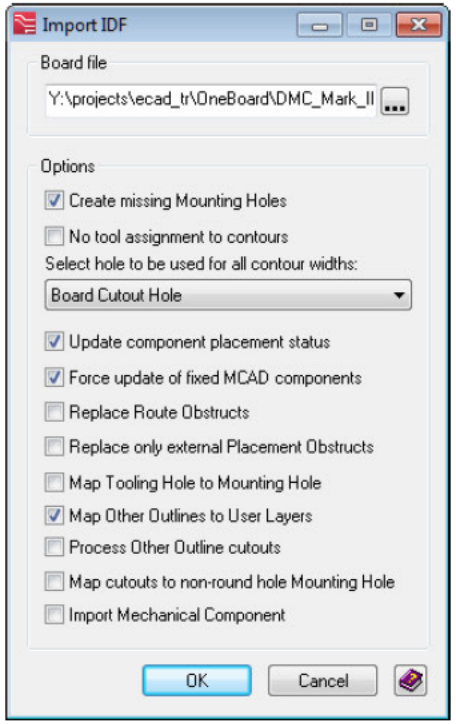


IDF Import to ECAD

- NGC > MCAD/ECAD > Import Mechanical IDF
  - Pulls the IDF file from Oneboard Storage
  - Runs custom PostProcessor in the background to preprocess the file
  - Invokes the Import IDF command with default options set

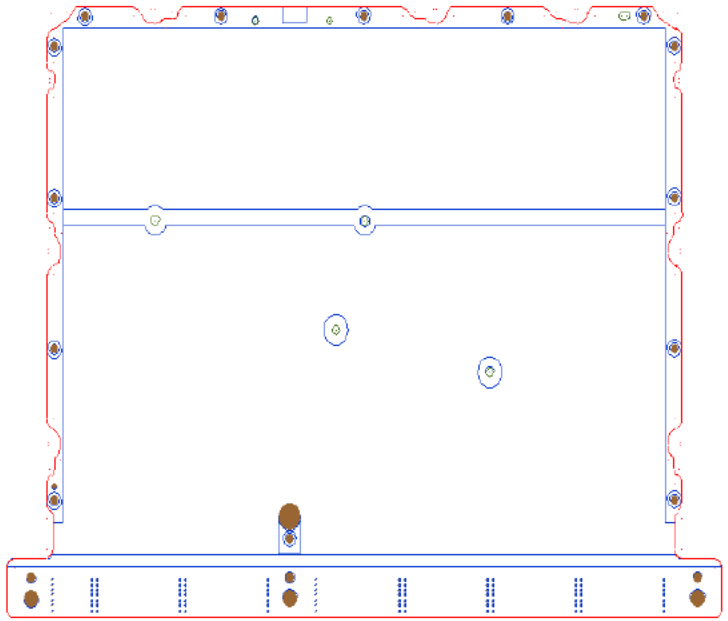






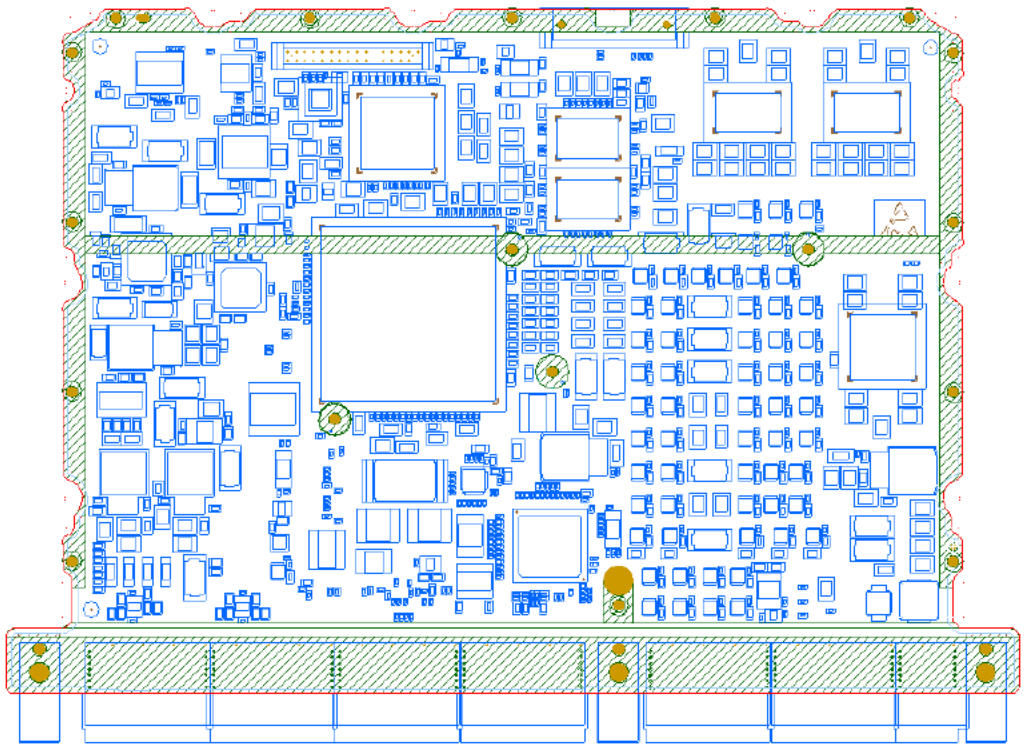
ECAD IDF Import Issues

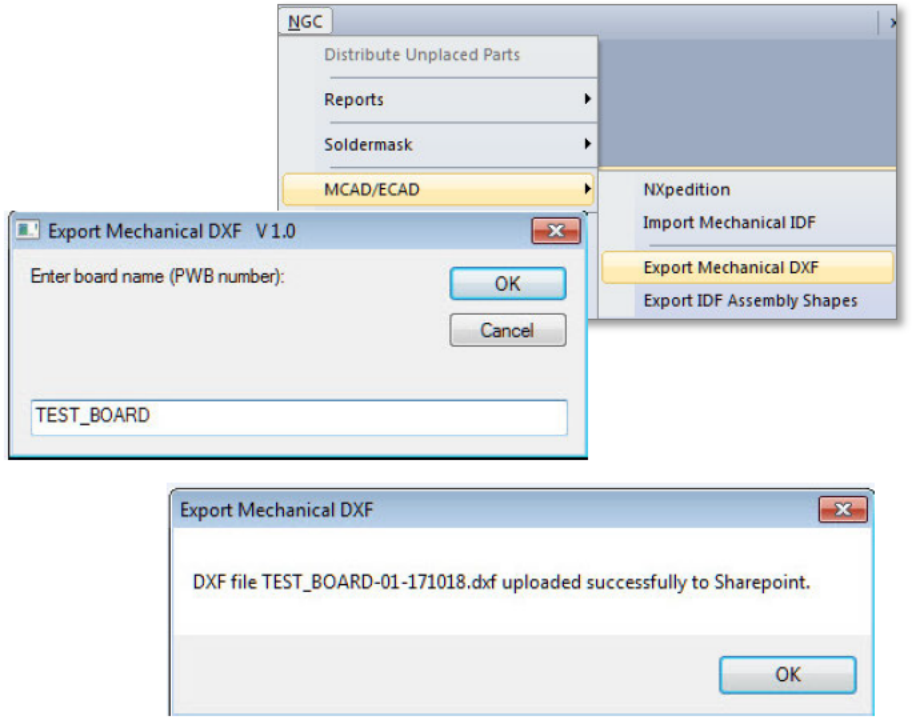
- Origin needs to be on an object, not a point in space
  - Different coordinate systems between MCAD and ECAD
  - Lower left corner of the board outline may be different location between the tools
  - Center of a mounting hole is good choice for origin
- Only one Board Outline can exist in the file
  - Slots/Cutouts should be defined as contours in MCAD
- Layers assigned to Other Outline shapes cannot have spaces in the name
  - Spaces will be converted to underscores by ECAD Import



DXF Export from ECAD

- NGC > MCAD/ECAD > Export Mechanical DXF
  - Generates a DXF file containing all part and copper objects on top and bottom layers of board
  - Writes the file to Oneboard storage



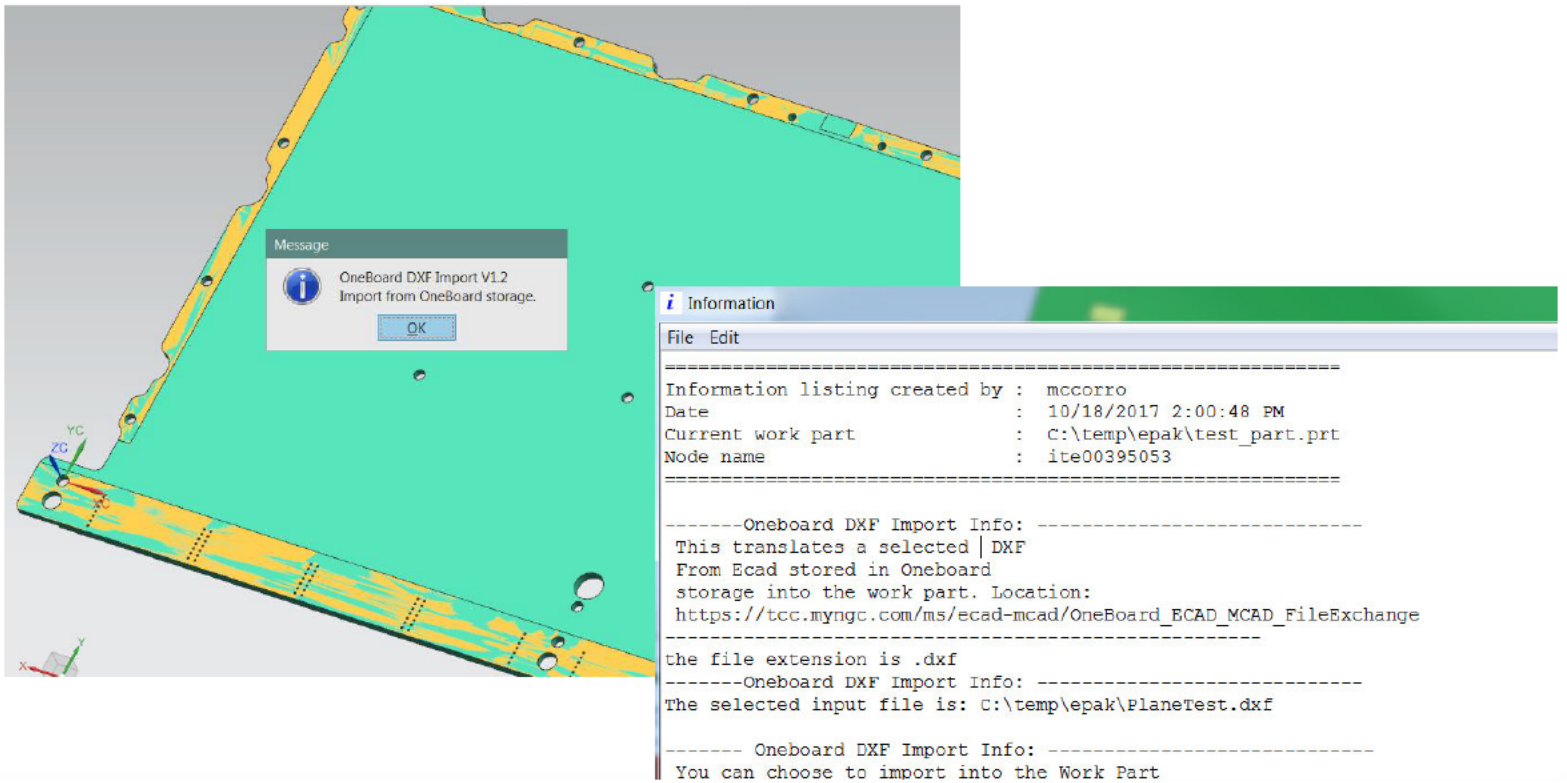


DXF Data Exported

- Board outline, origin, contours (cutouts), cavities
- Assembly outlines, reference designators
- Part pads, pin (plated) holes, and fiducials
- Mounting holes
- Routed traces, conductive shapes, planes, via pads, and via holes on top and bottom layers only
- IDF other outline layers from previously imported IDF files

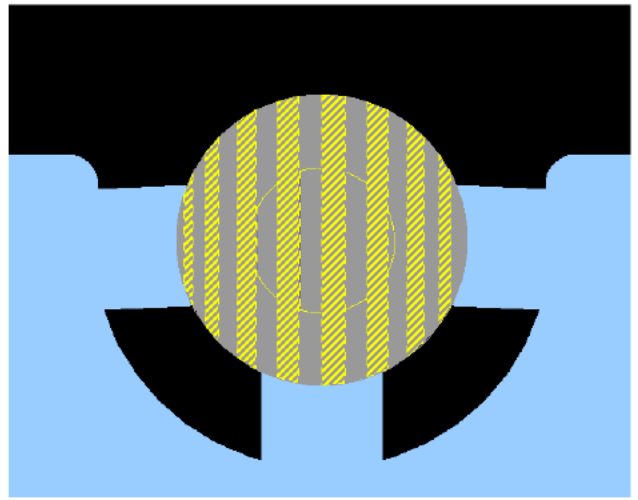
DXF Import To MCAD

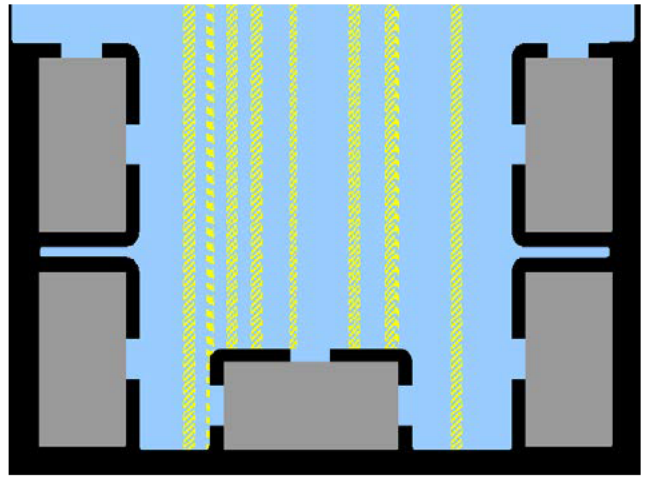
- Translators tab > Import DXF
  - *Pulls the file from Oneboard Storage*
  - *Imports the DXF file into the work part*



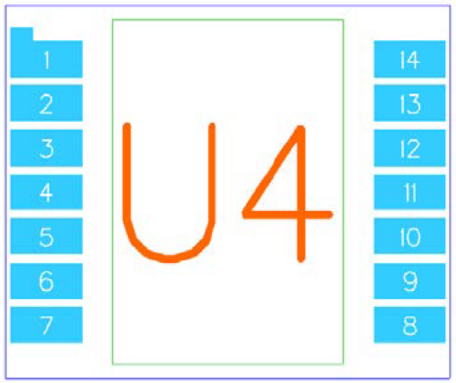
DXF Issues

- DXF data is not intelligent
  - *Dumb graphics only*
- DXF data can be very large
  - *Data is not contourized*
  - *Pads and planes not drawn as polygons*
  - *Filled planes converted to "many" lines*
  - *ECAD data stored on many layers*
- Incompatibilities in how ECAD and MCAD write and interpret DXF object types (Hatch, Fill, etc)
  - *Process refinement in process*
    - Automatically filter/exclude unneeded problem object types and layers.
    - Reduce file size and translation time





DXF versus IDF Export From ECAD



- Mechanicals prefer DXF to IDF for board placement data
- IDF files
  - Use placement outlines to represent parts
  - Placement outlines in ECAD library are large – 5 mils past the ends of the pads
  - Few MCAD models exist for electrical parts
  - Part height information included – parts with no MCAD models are shown in MCAD as blocks the size of the part outline, extruded the height of the part
  - Large placement outlines mean part blocks overlap each other in MCAD
  - No copper data included in IDF file
- DXF files
  - Use assembly outlines to represent parts
  - Assembly outlines in ECAD library represent true size of parts
  - No part height information included – all parts are shown in MCAD as flat polygons
  - DXF file includes copper features on the board

Why Use the File Management System we Selected?

- License cost was more for the competing product and we needed to add 100's of users to have licenses.
- ECAD users have no experience with the competing product
  - Big Install required
  - Need training for ECAD users
  - Need retraining every time new board starts
    - Electricals only work on design tools a few months each year
    - Spend remaining time simulating and testing designs
- Competing product automation not as streamlined
  - Our selected product process 1 step
  - Competing product many steps
- Competing product permissions
  - Electricals and Mechanicals need to be in same group to write to board item

IDF Electrical Process – Our Selected Product versus Competing Product

- Our selected product process
  1. *Open design on ECAD*
  2. *NGC > MCAD/ECAD > Import IDF*
  3. *Enter board number*
  4. *Click OK in Import IDF dialog box*

- Competing Product Process
  1. Open client
  2. Search for item
  3. Expand item and select IDF file
  4. Actions > Download File
  5. Select download location and download file
  6. Open design in ECAD
  7. File > Import > Import IDF
  8. Find and select IDF file
  9. Click OK in Import IDF dialog box

DXF Electrical Process – Our Selected Product vs Competing Product

- Our Selected Product process
  1. *Open design in ECAD*
  2. *NGC > MCAD/ECAD > Export DXF*
  3. *Enter board number*

- Competing Product process
  1. Open design in ECAD
  2. File > Export > Export DXF
  3. Select board layers to export
  4. Click OK in Export DXF dialog box
  5. Open Product client
  6. Search for item
  7. Create dataset under item
  8. Check-Out dataset
  9. Upload DXF file to dataset
  10. Check-In dataset

Implementation

- 3 months development in the ECAD, MCAD and file management environments
- 3 “shakeout” pilot project implementations
- Application rolled out to 800 MCAD users and 500 ECAD users
- Automations created through the use of MCAD Application Programing Interface (API) programming plus ECAD API programming
- Allows our mechanical and electrical engineers to focus on design rather than the mechanics of generating, transferring and controlling IDF and DXF files, thus trimming valuable time from the printed wiring board design process

Conclusion and Future Plans

- Reduced cycle time for file exchanges between ECAD and MCAD
- “One Button” imports and exports greatly reduce human errors such as output files having wrong version, file extension or units or import files bringing in unnecessary layers and objects.
- Eliminates manual version tracking and location of files
- Automatic Post Processing compensates for deficiencies of the ECAD and MCAD import translators
- Future Plans:
  - Continuous Improvement through better filtering of DXF to MCAD (file size)
  - Better predictive behavior of outlines in MCAD to IDF (top vs bottom)
  - Investigate migration of file storage system into competing product already in use to manage MCAD files
- Merger of the MCAD and ECAD companies – unknown effects

Acknowledgements

- Mike Rindos, Lead
- Kelli Hosier, EEDS Tool Support
- Steve Krug, MCAD superuser
- Chad Lee, ECAD superuser
- Robb McCord, EEDS Tool Support
- Doug Renwick, EEDS Tool Support
- Helen Park, Management Sponsor
- Karen McConnell, Management Sponsor

OneBoard CompleteTeam

Questions?

Thank you