

must comply with certain data integrity rules. The Lock Check essentially ensures that the file is not prohibited from entering the library due to an existing overlay or processing lock. It then checks to see if the file is "owned" by anyone. If so, it must be the user or someone for whom the user is a surrogate. In the latter case, the lock would be reset in favor of the user, and notification would be sent to the original owner. Once the Lock Check is cleared, Step 39424, Inst List is executed. Here the file is added to a list of files to be installed into the library at the end of the processing. The last thing the program must do with an "M" record is Add to BOM in Step 39425. In the preferred embodiment, this is done by parsing the file information from the second and third tokens into the proper format to write it into a temporary file. This file is used in a later by the Aggregation Manager's API which creates BOMs for the DMS. The fourth token, which is the BOM Member Type, is also written along with the file information.

Step 39426 in FIG. 22c, R Rec, determines if the line is a Result record denoted by an "R" as the first token. If so, Step 39427 Find Proc is executed. In this Step the third token is parsed into the library filetype. The Target Package, Level and Version contained in the control information transmitted from Step 39324 is used as the PFVL. This information is used to filter the search of all processes in the DMS for that package. The process names are compared against the second token. If a match is found, Step 39428, Crit Chk, is run. Here the criteria for that process is compared against the actual process result in the fifth token. If the result meets the criteria, Step 39429, Rslt List is performed. All information required to store the result is saved for a future Step. This information contains the process name, version, library filetype, Target Level and the result.

Step 39430, P Rec, determines if the line is the Master Pedigree Record denoted by a "P" as the first token. If so, Step 39431 Spec FIC is invoked to perform a special CRC check. The purpose of this check is to ensure that the actual PED file being processed is identical to the one generated by the task in Step 39101 of FIG. 20. In our preferred embodiment, the "P" record is always last, so all records excluding the last one are temporarily stored into a file. The CRC routine is run on this file and the result should match the fourth token of the Master Pedigree Record. If not, then the PED file has been altered since creation and the contents can't be used. If this check passes, the temporary file is discarded, and Step 39432 Chk FN is executed. Here the second token of the Master Pedigree Record is compared against the actual file name being processed. This ensures the contents of two PED files weren't swapped. The final check, Chk FT, is Step 39433. This simply ensures that the library filetype denoted by the third token is indeed that which the DMS expects for External Data Control.

Step 39434, K Rec, determines if the line is a Support record denoted by a "K" as the first token. The files identified by the second token in these records only need to be run through the same Basic FIC check as in Step 39421.

At this point the program returns to the top of the Rec Loop in Step 39415 and repeats Steps 39416 through 39434 until all records of the current PED have been examined. Once the loop is exhausted, control returns to the top of the PED Loop in Step 39412. Once again, all steps from 39411 through 39434 are repeated until all PED files are examined.

Once all PED files have been successfully processed, it's safe to continue with the steps that actually store data in the repository and set results in the DMS. Step 39436, Inst Files, is invoked to deposit the files into the repository. All information required for this task was saved during Step

39424. In our preferred embodiment, this step relies on our Install Algorithm to store the data into the DMS. It should be noted that there is no need to actually save the PED file itself since it is merely a conduit by which other files and results are transported.

Step 39437, Make BOM is initiated to use the Bill of Material information gathered in Step 39425 to actually create the BOM in the DMS. This is done via the Application Program Interface provided by our Aggregation Manager. In Step 39438, Rslt, all the result information gathered in Step 39429 is stored into the DMS. This is done via various functions provided by the Process Manager for setting process results. These are described in the previous section. The program finishes by executing Step 39439, EC Cntl which checks to see if any Problem Fix, Part Number or Engineering Change information was transported with the PED. If so, the information is loaded into the proper tables of the DMS.

Upon completion of the above steps, the data and all results used and/or generated in Step 39101 of FIG. 20 is safely stored into the DMS in a manner consistent with the rules regarding Automated Library Processing.

While we have described our preferred embodiments of our inventions it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the inventions first disclosed.

What is claimed is:

1. A data management system for file and database management useful in concurrent engineering processes, comprising:

a design control system for fulfilling requests of a user initiated from a computer system client system coupled to a network, including

a data management control system for managing a plurality of projects, each project having a data repository for data records and a control repository comprising a common access interface and one or more databases,

said control repository communicating with users of said design control system for fulfilling requests of a user and the data repositories of said data management control system through a plurality of managers, each manager performing a unique function, wherein said managers act as building blocks which can be combined in a plurality of manners to support an environment for suitable for multiple users of a user community, and

an external data control environment coupled to said data management control system via said common access interface for transferring results and data into said data management control system for storage in said data control repository from a process executed outside of said data management control system; and wherein

one of said managers is a process manager, and one of said managers is a library manager enabling automated library processing of any application program or tool to be launched using input data from said data management control system and after processing enabling results to be recorded as output data in any data control repository of said data management control system as a record in said data control repository;

wherein each of said records in said data control repository includes data indicating the record filename, library, filetype, version and level; and,