




Towards an extended model-based definition for the digital twin

Alexander McDermott Miller , Ramon Alvarez  and Nathan Hartman 

Purdue University, USA

ABSTRACT

The concept of the digital twin calls for virtual replicas of real world products. Achieving this requires a sophisticated network of models that have a level of interconnectivity. The authors attempted to improve model interconnectivity by enhancing the computer-aided design model with spatially related non-geometric data. A tool was created to store, visualize, and search for spatial data within the computer-aided design tool. This enables both model authors, and consumers to utilize information inside the CAD tool which traditionally would have existed in separate software.

KEYWORDS

Extending MBD; digital twin; CAD

1. Introduction

Model-based definition (MBD) is an emerging paradigm which utilizes 3D-CAD models to carry engineering information. The introduction of CAD tools led to the realization that drawings may no longer be the best method to communicate information. The 3D format offers significant advantages over its 2D counterpart. In addition, adoption of MBD offers significant cost and time savings [14]. One account found that “a savings of more than \$3M annually at a single naval facility will be generated by implementing MBD” along with “a 33 percent reduction in the development schedule of new items is expected” and “MBD shall also create a significant reduction in manufacturing errors, which should result in a decrease in the amount of rework, providing significant cost savings.” [18]. Moving forward, model-based definition efforts will be focused on enhancing the model-based definition capabilities beyond drawing information; and this exaptation will likely lead to many unforeseen benefits and opportunities.

The original focus of MBD concerned the communication channel between design and manufacturing, but now it is spreading across the entire organization. Model-based definition no longer refers solely to design or manufacturing; it is not solely a geometric activity. Modern model-based definition theory encompasses all aspects of a product. Model-based definition focuses on how a product is a composition of multiple models. A single product is comprised of several distinct aspects; each aspect, therefore, needs its own model [1, 3, 12]. As a result, a single product contains multiple models. This

definition is effective, but what potential is there in their integration? The integration of the multiple models of a model-based enterprise is the digital twin. “A digital twin is an integrated multi-physics, multi-scale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin” [5]. The digital twin is a perfect replica of a physical object but it is digital, not physical [5, 15, 18]. This idea parallels the work related to digital product memory or semantic product memory. Each physical object has a digital replica which carries information that describes a single instance of the digital twin [8]. It achieves this replication by integrating multiple models together to create a better image of the physical world. Although the long-term goal for MBD is the digital twin, the path to create the necessary model-based enterprise for the digital twin is unclear. This research explored this problem by attempting to integrate behavioral attributes with the shape by the storing of behavioral information within the computer-aided design model. Current model sharing practices do not communicate adequate information because, as of now, a CAD model only carries a small portion of the total product information [11].

2. Background

2.1. Model-based definition - behavior

Model-based definition adoption objectives which originated in the 1980s and 90s have begun to be realized by model-based definition leaders. Models have assumed

fully the position that drawings had previously held [14]. Considering the success of model-based definition, the time has come to expand MBD beyond drawing related information. A possible next step would be to develop the roadmap for the digital twin. The digital twin paradigm garners wide support due to the potential benefits and improvements such as automation and intelligence. Although the digital twin is an appealing concept, it lacks formality. Creating a digital twin has numerous challenges. For example, there is no road map for its implementation, and it is also not clear what information or models need to be integrated. These are major challenges for the development of the digital twin. However, what is understood is that the digital twin is a composition of disparate digital models which gives rise to a higher fidelity model of a product. Therefore, this research focused on the following: *which* models need to be integrated, and *how* to integrate them into the digital twin. These two questions were explored through the integration of behavior and geometry in the computer-aided design model [16].

Behavior does not refer to the performance of a product, but instead refers to the physical manifestations that occur in a specific part in response to external stimuli [4]. The external stimuli can hypothetically, be anything, but they will most often be restricted to other parts in the system, as well as the operating environment. Behavior is an output of a system of relationships between an object's physical characteristics and its external stimuli. The material, material structure, and shape are some examples of physical characteristics that affect behavior. Due to the high complexity of behavior, there have been several research efforts that have attempted to analyze a product's behavior [4, 9, 11, 13]. One model of behavior emphasizes the interaction of structure, function and behavior [4]. Another model defines it in terms of behavior and structure, but emphasizes the relationship of external stimuli to a changing behavior state [17]. These two models support the theory that behavior emerges from the physical structure; it is behavior, in effect, which creates the function.

Historically, however, the structure has defined a product. *Structure* is the physical characteristics of a part [4]. It includes shape, material, and all other attributes which communicate its physical characteristics. This is arguably the most understood aspect of an object. Traditionally, a structure would be prototyped and tested in order to determine its behavior. Once a structure's behavior was certified, it was considered "complete." The structure would be designed to deliver a desired behavior, and this behavior was analyzed to determine if it met the functional requirements. Desirable behaviors, achieved through the physical structure and external stimuli, are

functions [4]. These functions ultimately drive our physical structure even though they are a byproduct of the physical structure by means of behavior. It appears to be counterintuitive that the behavior drives the product's purpose, but its use is for verification rather than design. This is because our most effective design methods were structurally based. Without modern computing power, we could not efficiently predict behavior. It was necessary to build the structure and physically stress it. Model-based definition and the digital twin provides a new opportunity [11]. Using computer simulation and modeling, we are now capable of replicating the world at a level never before seen. The digital twin calls for the integration of multiple models into one master definition, and maybe it is then possible to change altogether how a product is designed. Researchers have already begun to find success in creating alternate structures by integrating behavior into structural design processes [13]. One example integrates behavior more readily into the design process to allow for more creativity. It lessens the control that structure currently has over our design processes and thereby creating the opportunities to find more effective designs [13]. Another method is the specification of a behavior for a structure. Designers would manually input the purpose behind a structure so that downstream information consumers would better understand the purpose behind the structure. This altogether avoided the integration of multiple models, but it did enrich the structural definition by allowing for the communication of behavior [19]. Moving forward in model-based definition requires the integration of behavior and shape so that the digital twin can become a reality.

Defining a function based on the behavior is challenging, because we understand relatively poorly the link—behavior—between structure and function [11]. It is not easy to identify the behavior of a given structure without some analysis tool, and even with analysis there are multiple behaviors which contribute to a functional part. Because our verification of behavior is based off the structure, it became the ultimate test for capabilities. Although the definition of a product is captured and verified through the structure, it may be possible to do this through behavior, because ultimately it is the behavior that drives a function.

Products are designed to be specific shapes. These shapes ultimately become how they are validated for performance. The shape becomes validation specifications, how a part is evaluated against its functional specifications. Current processes use geometry as the validation specifications.

Once an object was manufactured, inspection checked its shape to ensure it fell within the validated specifications. Humans would constrain an object down to a

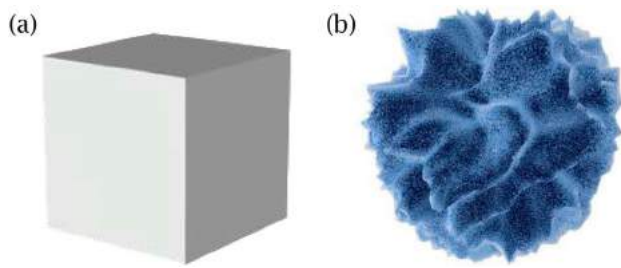


Figure 1. Model of the acceptable variation of a product. (a) Shape specifications and (b) functional specifications.

specific and tested shape, because our methods for analysis were too expensive for any other method. This led to the wide adoption of shape specifications, a highly constrained and understood shape and material that could deliver the desirable behavior. Shape specifications, Fig. 1(a), are the possible variations of a product within a specific geometric definition which meet the prerequisite performance parameters. This is the current level of validation specification. However, there is a vast array of variation which can still meet the necessary functional objectives of the product. These variations have simply not been evaluated.

All possible variations of a product definition that accomplish all performance parameters, even the geometry configurations that are not used, are functional specifications. An image of this can be seen in Fig. 1(b). This image represents that the possible ways to achieve a function are highly variable, and the specific constraints that are placed on a product are not the only shape that could work. The shape specification is always a subset of functional specifications. The relationship between Fig. 1(a) to Fig. 1(b) can be seen in Fig. 2(a). A concrete example to support these concepts can be understood through airplane designs. As a team decides on the performance parameters of the plane they shape the functional specifications, Fig. 1(b). The amount of lift, the speed, or mission range. These don't select specifics necessarily but they shape what is possible. Shape specifications, Fig. 1(a), select specific geometries. The wing span, curvature of the wings, or fuel tank. There are feasible designs which are not selected within the functional specifications because it was not cost efficient to test all manufacturing variations. If the wing span came out larger then validated, or smaller, then the part would either need to be validated or scrapped. This can be applied to any component within the plane. Something as simple as a nut that measured outside Fig. 1(a), but sits within Fig. 1(b) would require validation or it was scrapped. The digital twin is reaching towards the functional specification instead of shape specification, because it is becoming more cost efficient

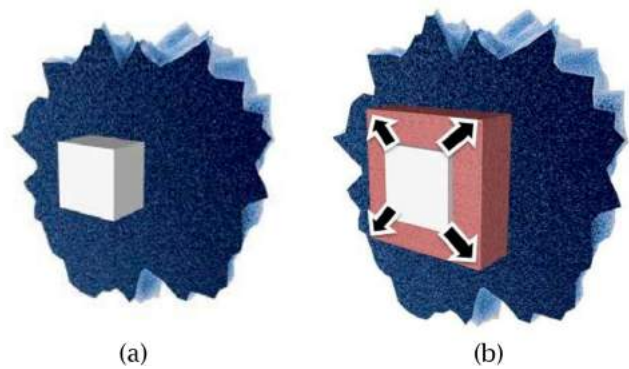


Figure 2. Shape specifications as a subset of functional specifications, and the expansion of shape specifications utilizing behavior. (a) Shape specifications within functional, and (b) expansion of shape specifications using behavior.

to evaluate a product for its functional specification instead.

By incorporating behavioral information into the shape specification, it expands the validation specification, as can be seen in Fig. 2(b). What constitutes a functional part, is not necessarily a shape; therefore, the definition of a product should evolve to better fit with our capabilities. However, a major challenge with leveraging the behavioral information is that multiple behaviors ultimately contribute to functionality. Utilizing multiple behavioral models without bias is an important aspect of design verification through behavior [4]. This research sought to expand the tools for definition to leverage this understanding of a product.

Current PLM tools can store information in separate forms within a database, but this does not build the relationship needed for a digital twin. They exist as separate entities categorized together instead of creating a more sophisticated definition. This problem was also explored via knowledge-based representations, and focused on creating associations between the different models. The researchers documented the difficulty of associating non-geometric information with geometric information, and the corresponding difficulties which arise due to unidentified associations [5]. The goal of this research was to build associations between behavior and shape through the 3D-CAD model, but to do this the CAD model needed extending. Therefore, the computer-aided design tool was enhanced to accommodate for the storage and retrieval of behavioral information in the 3D-CAD model. This leads to a second important question, what behavioral information can and should be stored in the CAD model. This question is best answered by those implementing an MBD implementation, as the needs and information vary between industries. This research did

select a specific type of information with which to use in storage.

The research focused on spatially indexed information. This refers to characteristics of the object that will vary across the surface or throughout the volume of the object. An example of this would be how the surface of an object will never be perfectly flat using today's manufacturing methods. This difference between nominal topology and actual topology affects the behavior of the part. It is not only geometric information, but also could be physical characteristics such as hardness or residual stress. This research effort sought to integrate behavior and into the CAD model by enriching the model-based definition with behavioral data. The behavioral datasets are stored inside the CAD model in the locations that they reference. The CAD model acts as an index for the behavioral datasets. This tool could provide a significant advantage to those exploring the CAD model and begin the necessary integration for the digital twin. Normally separate models would be registered together by embedding the data generated, by separate models, in the computer-aided design model. This work represents an effort to bring MBD to a more refined state, and chart the path towards the digital twin goal. This research set forth to explore possible methods for storing and retrieving information from within the CAD model. It led to the development of a software which is capable of storage and retrieval of information in the 3D-CAD model.

2.2. Model-based definition – tools and methodologies

Model-based definition's maturation has spurred disparate research efforts to improve a product's definition. Cutting edge technologies enable new forms of product definition. These new forms vary from definition presentation to alternative definition methods. This section will provide details to paint the landscape of MBD research, and place this technology within it.

A product definition is the method by which humans structure and understand information which defines an object. The overall product definition is the sum of a multitude of domain-specific definitions. Each domain has its own methods for defining their aspects of the product definition. For example in early MBD efforts, the model solely represented the geometric properties. While physics models would exist separately and define behavioral attributes. This was done because it was the most effective method we had. However, technology is changing our abilities to define products, and enables the merging of product definitions into a more holistic form. Researchers have begun exploring a multiple viewpoint

model, which accounts for the numerous stakeholders, or workflows within an organization [1,3]. One example research effort in 2017 from Oamer-ali et al. emphasized the need for presenting information based on the different perspectives within the organization. This framework improves the usability of data by better managing what information the different facets of an organization need [12]. What has yet to be decided is how these different domain definitions interact, fit together, and carry information.

Digital product memory or semantic product memory refers to the sensing and capture of information related to specific physical part [8, 15]. Digital product memory refers to the one to one aspect of a digital twin instantiation. Each physical object has a digital replica, and information which contributes to understanding of the physical product is carried within. The research presented here supports the capture of digital product memory, but does so by utilizing the computer-aided design model as the repository of information.

The extension of the CAD models abilities to carry information has been of recent interest to researchers. Approaches to capture this information have differed greatly, but often they focus on integrating information into the CAD model which normally would be excluded or left out. One effort enabled the model annotations to carry information related to design intent. This enhances the CAD model by carrying information which helps describe the reasoning behind decisions that defined the product [2]. Another tool focuses on improving the communication channel between design and manufacturing by creating an ontology [6]. Similar to these tools, most extensions to the CAD tool rest within the capture and communication of information in design. Most often when this information is communicated towards manufacturing. However integration of different workflows into the MBD are needed to move towards a digital twin. As stated by Hedberg et al., "How is information captured effectively and made available for designers to evaluate DfX opportunities (e.g. manufacturability, producibility, sustainability and other 'ilities') earlier in the design cycle? We expect this to be a difficult question to answer because capturing the information happens in various non-standardized ways throughout industry. Enterprises have developed proprietary data capture processes, which they treat as trade secrets and competitive advantage [7]." The tool developed here provides a method for data capture within the computer-aided design model. The novelty of this tools emerges from the integration of datasets which are normally excluded from the model-based definition. Furthermore, they can now be cross-referenced within a geometric setting.

3. Implementation

3.1. Implementation outline

In the initial phases of the project many different concepts of the MBD philosophy were taken into account. The goal of this project is to expand the MBD framework and attempt to not make exclusions of the concepts of the MBD framework. For our software solution, the development cycles of the project were centered on the philosophy of the MBD which is agnostic from any specific software or platform. This greatly influenced the software solution to use concepts and implementation methodologies which could be implanted in other software packages, not just the one the plug-in was originally developed in. A decision was made to create a plug-in to handle the interactions we wanted to have with the CAD software. The project went through a continual process of additions and modifications to the Software Architecture which would be followed with software development and user experience (UX) testing.

3.2. Software architecture

As part of the initial development cycles, a software architecture diagram, see Fig. 3, was created in order to better visualize what functions would be created in the plug-in. In order to keep our software solution faithful to the agnostic MBD philosophy the plug-in was developed with the most minimal use brand specific software tools. Much like MBD is not dependent on one particular product or one particular enterprise size so too should the plug-in be built so that it does not solely rely on one product and to be scalable for any enterprise needs. While it would be possible to create a program which can

externally combine data and a model from a CAD package it was determined to be easier for validation reasons to create a plug-in which would function with the CAD package and embed the data within the CAD package. In this solution there will not be a need to validate the accuracy of the CAD model being used but instead all that will be needed is to validate the data that the plug-in is connecting to the model.

The research explored multiple software architecture ideas and implementation methods in order to weigh the benefits and determinants that could affect the results of the expanded MBD CAD data. In this research, methodologies were selected for the software architecture which would minimize the dependencies of brand specific CAD software. While this would create more programming overhead in order to implement a solution it would in the long run allow the developed software, which implements the expanded MBD solution, to be more resilient and less prone to failures from future brand specific CAD software upgrades or changes. This chosen methodology would also allow future version of the software architecture to be ported over to other CAD software brands or to be developed to run independently of a CAD environment, such as allowing PLM software or other independent software packages to interact with the data.

Many CAD packages have software development kits (SDK) which aid in the creation of plug-ins. However, many of those SDKs have what we consider to be “black box” utilities or tools. These are functions, which can be implemented in plug-ins or user scripts, can do different tasks with data, such as process “data A” to create “data Z”. However, these “black box” functions do not allow the programmer to have any degree of control over the calculations which happen in-between the input and output of

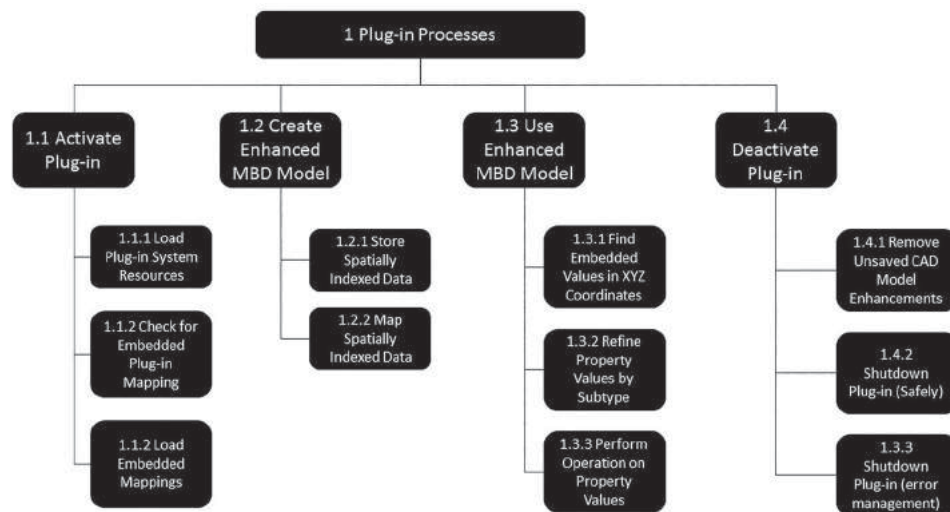


Figure 3. Functional architecture diagram.

data. It is also common for those “black box” functions to not allow for the capture of the different in-between processed data intermediates. A major issue with these “black box” functions is that they tend to be specific to the CAD package; for example, CAD program A may offer a specific tool that CAD program B does not offer or use.

Creating any plug-in will require the use of brand specific functions in order to connect the plug-in with a specific CAD package. The end goal is to minimize the dependency of those CAD specific packages. By minimizing the reliance on specific CAD package “black box” functions a software architecture arises where the main reliance of those “black box” functions is kept primary in activating the plug in, such as in Fig. 3 section 1.1, and in safely shutting down the plug-in, such as in Fig. 3 section 1.4. This creates an environment where the core programming of the plug-in can mostly remain unmodified and all that would be needed is to swap out the initiation and shutdown methodologies implemented that connect the core programming to a specific CAD package. Also the data processing functions of the plug-in would be easier to transfer out to a standalone CAD package independent program that could be built to take in a neutral file formats containing the expanded MBD CAD model.

The creation and enhancement of the 3D-Model is a main function of the plug-in as can be seen in Fig. 3 Section 1.2. This demonstrates where the plug-in takes in external datasets, such as data located in excel or text files, and brings that information into the CAD package. The data is embedded into the CAD model or the surroundings of the CAD model. Another option for data input, is using the CAD model as a 3D index. The user could click a location in the computer-aided design model, and manually enter in information. Then the information is embedded into the model. The user has the ability to manually embed data at a specific location on the model surface or within the model, as in embed data millimeters or inches within the model. Or the user can add information off of the model in the surrounding area, such as if the user wanted to add information an inch above the model. The current implementation of the plug-in uses this method of data embedding so that information can be embedded at the highest precision level that the native CAD package can handle. This allows data to be embedded on large features such as face or on fine small objects such as the scale of an individual vertex.

The current methodology embeds data in the native CAD package as an object, which will be referred to as Attribute Marker. This means that the data can be searched for without the need of the plug-in in cases of extremely small data sets. The embedded data uses a simple schema, location in an xyz format an attribute name and its value. A given xyz location could hold multiple

Attribute Markers each holding only one piece of information, such as residual stress equaling a specific amount, or the Attribute Marker could hold multiple pieces of information at that given xyz location, such as residual stress, clearance value, surface roughness, hardness. In the case of multiple attributes being stored at a given location there would be only one xyz but then a series of attribute value attribute value sequences.

The kinds of data that are possible to embed via the plug-in currently include text and numbers. Following the schema described above, of xyz location followed by user defined Attribute Marker and finally numeric value, will result in the ability for the user to query the CAD model for any Attribute Marker or patricidal wording to similar Attribute Markers. As an example if a user wants to create a list of “manufacturing stress” points to embed into the CAD model then the user can generate a csv or Excel worksheet containing “[xyz numeric location], manufacturing stress, [numeric representation of value]”. It is important to note that the user could have a single dataset to import with mixed Attribute Markers, such as “manufacturing stress” and “stress of in field use”. The schema would be the same even though the user is combining multiple datasets into one file prior to embedding data, “[xyz numeric location],[name of a single Attribute Marker],[numeric representation of value]”. Users will be able to search for keywords, as explained in the Software Overview bellow, which would allow the user to either search just for “manufacturing stress” or for “stress”, which would provide data for both the Attribute Markers mentioned. The limitations in the xyz location currently make it such that the information must be three integers separated by a comma, one comma separated value for each of the planes. The numeric representation can be expressed by the user as any rational number. It is important to state that there are no preprogrammed Attribute Markers in the plug-in. All Attribute Marker names are determined by the user. The limitations in the Attribute Marker are currently that there are reserved characters, by the CAD package which we are implementing our plug-in in, which cannot be used as well as the fact that we have reserved the use of the comma. The Attribute Marker naming schema is following regular expression, regex, of alphanumeric and spaces. This schema was not expanded to include more ASCII characters to avoid issues where different CAD packages have reserved different sets of characters. The limitation in the value for the Attribute Marker is that it must be a numeric representation

The data once embedded in the CAD model can either be saved in that CAD package’s native format or it can be saved out to a neutral file format. Compatibility of the embedded data in the CAD package is exactly the

same as how that given CAD package handles forward and backward compatibility between files. If CAD package A has an old and new version and the old version will not allow saved files from the new version to be imported into the old version then data that is embedded in the new version native CAD package format will not be imported into the old version. If the old version allows files from the new version to be imported into the old version then the embedded data in the new version can also be imported in. In the case of the first issue where there is only a forward compatibility between native saved files in the CAD package an alternative to getting data that was created in the new version into the old version can use a native file format (NFF) to import in the model and embedded data. The NFF format used by the plug-in is the STEP format. The compatibility of the emended data being imported into a given CAD package is dependent on that CAD package's import system being built to follow the standards as outlined by the STEP format.

The user also has the ability to decide what kind of visualization or reporting method is best suited to display the embedded data, which are planned in Fig. 3 section 1.3. The desired information could be displayed directly onto the model in several preprogrammed methods, such as colored locations or have the data displayed as text right on the model much like PMI information would be displayed. Also the information can be retrieved and displayed as text in a reporting window if no visualization method is desired.

As an example, if a User has several datasets such as performance and manufacturing data concerning a wing. Those datasets could be embedded into a wing model and the user can bring up all the information concerning the wing stress or the wing wear and display the locations of where that data is located on the CAD model as well as obtain the information in text format. This would be done all within the CAD package that the user already has experience with. The user could also embed additional data sets which include attributes about that model, such as wear rate and residual stress from manufacturing, and be able to save and share that file with colleagues.

The user's colleague could then use the software solution to see what is already embedded in the model with simple exploration tools, such as a drop-down box which shows data from attributes embedded into that model, and use the software solution to automatically create a visualization to better see the desired data.

3.3. Software overview

The plug-in was created to work with the API of a specific CAD package but the architecture of the plug-in was developed so that API differences between CAD packages could be swapped out in order to allow this plug-in to be brought to other CAD packages in the future. At the current time the plug-in itself can only read csv, comma separated value, or Excel files. The plug-in itself is incapable of reading CAD files and it is only able to interact with CAD data by running within a CAD package.

In order for a user to embed data into a desired model the user must load the plug-in into the same environment with the desired part file. In general it is assumed that users will have large amounts of data saved and so the user can use the automated data import option, Fig. 4 (a), of the plug-in to bring in data found in text, excel or other formats. The user also has the ability to use the 3D CAD model as a physical clickable index, Fig. 4 (b), to embed data on a specific surface location. If the user wishes to embed data manually with in the model instead of on its surface the user can use the specific coordinate embedding method, Fig. 4 (c).

Under normal operation the plug-in will embed the data into the model and the model will look like its original unmodified state, Fig. 5 (a). The user will receive notification from the plug-in when the embedding process has been completed and from there the user can choose different visualization methods to aid in understanding of the data embedded such as Fig. 5 (b) which shows the location of embedded data by displaying markers of where the data is located.

After data has been embedded into the model, or a CAD model is opened which already has data embedded

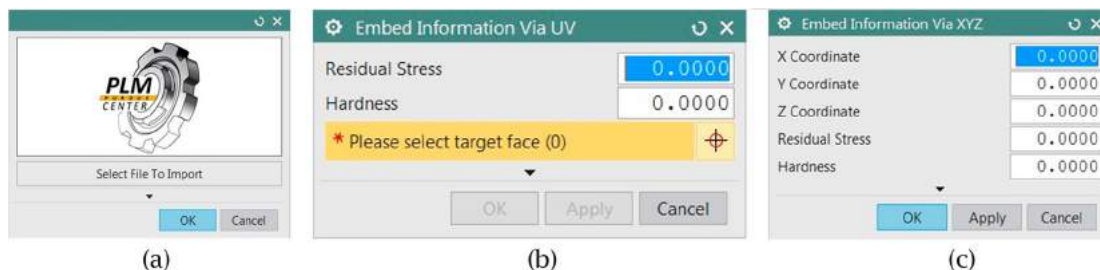


Figure 4. Embedding methods available in the plug-in. (a) Automated data embedding method, (b) Manual data embedding using 3D model as interactive index, and (c) Manual data embedding using specific coordinates.

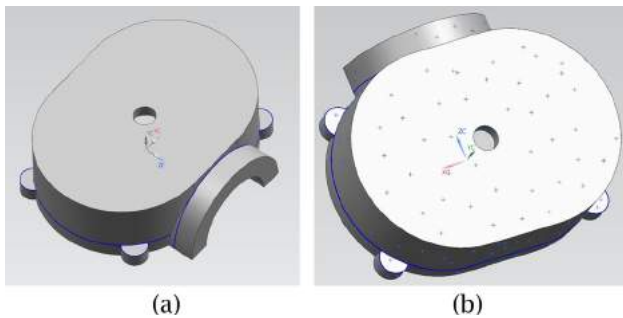


Figure 5. CAD model pre and post plug-in use. (a) A CAD model before having data embedded into it, (b) The same model after having data embedded into it and with a visualization method applied to the data.

in it, the user can load the Data Exploration window of the plug in, Fig. 6 (a). From within this plug-in window the user will have several tools to search for data embedded in the model as well as select different visualization, reporting and data computational methods. The Data Exploration window has the ability for the user to type in search words for the plug-in to find or the user can use the plug-in's automated keyword boxes which generates property information from the embedded data found in the model, as shown in Fig. 6 (b). When a user types in search words the system looks for those keywords in the embedded Attribute Markers. It is possible for the user to search for partial keywords. In the case that the model is embedded with data concerning "Surface Roughness" and "Surface Wear" a user could search for "Surface" and the plug in would provide details for both types of Attribute Markers. The plug-in primarily uses a window to display text, Fig. 6(c), to communicate with the user. This method is used to alert the user of tasks

that the plug-in is processing, it displays completed tasks, reports errors found, suggests solutions to errors as well outputs desired data on screen in a simple to read format.

A user can easily see data of interest by using the Data Exploration window and selecting a simple visualization option such as highlighting the data in the desired range with a desired color, such as the example shown in Fig. 7(a) which could illustrate the location of a crack on a CAD model. The user can also combine different visualizations techniques such as in Fig. 7(b) where a user could load the wear rates that they have mapped of a given product and have that information displayed on the CAD model. The use of tags to see where the data exists may not reveal much to a user however a combination of visualization methods may provide a better in sight. In the example bellow in Fig. 7(b) a user has colors to indicate when the value of a given property, in this case part wear, has passed a given acceptable range as well as have that data create spheres that grow in size to indicate how large that value is.

Once the user is done using the plug-in, the user saves any desired work or changes to the file using the same method that is normally used by the CAD package, such as saving in to an NFF or via the native format provided by the CAD package. The plug-in has shutdown detections mechanisms to make sure it safely shuts down. Shutting down of the CAD package or of the plug-in window will trigger the internal shutdown mechanism of the plug-in.

In the example shown in Fig. 7 only one attribute, data for wear rate, is being explored and visualized but a possible use case could be that the user embeds different types of datasets containing any quantity of attributes into the CAD model and simultaneously visualize all the

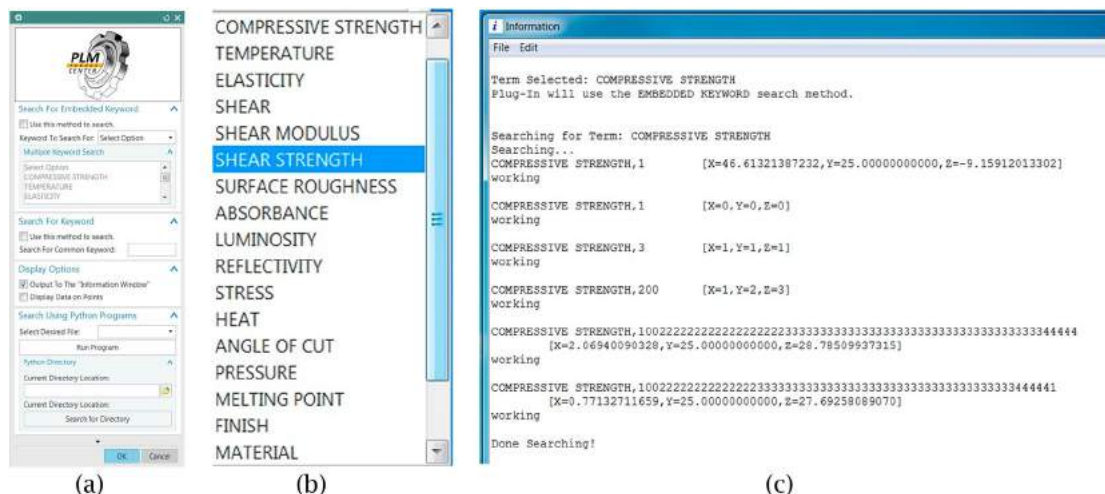


Figure 6. Data Exploration options available in the plug-in. (a) The main search engine with multiple data exploration methods as well as data manipulation and visualization options, (b) A close up of the plug-in's automated keyword search tool, (c) One of the basic data visualization methods.

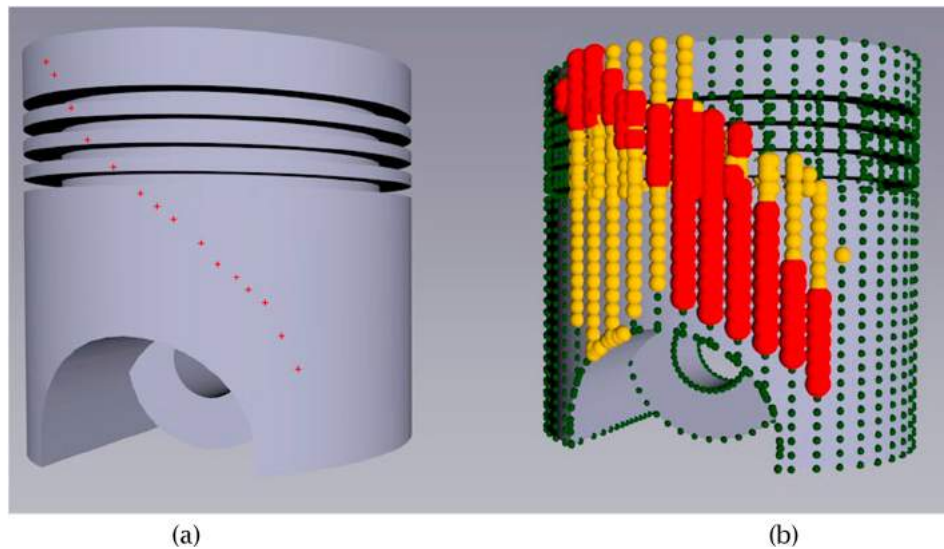


Figure 7. Combined visualization method exploring the property values of wear rate. (a) This visualization method changes the color of the data markers a user is interested in viewing. (b) The visualization technique takes into account a rate of wear and then assigns a color and shape determined by the value in that property.

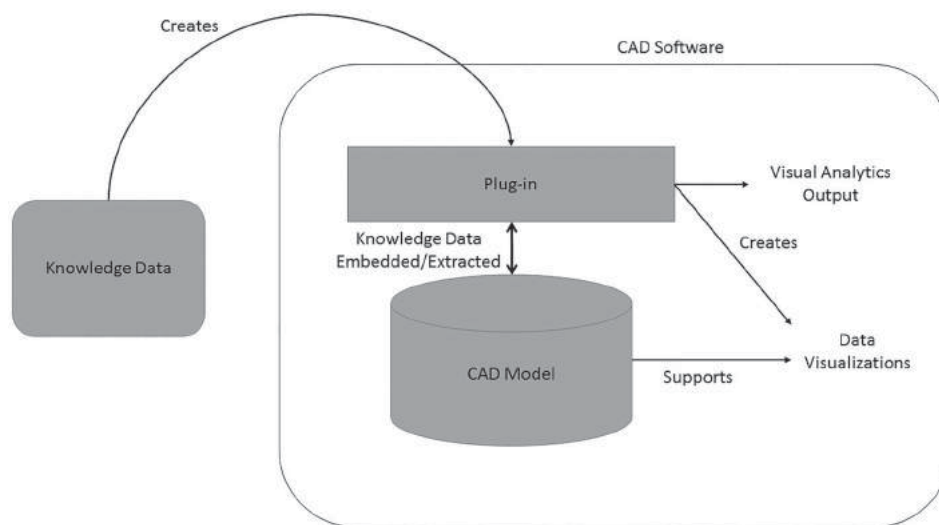


Figure 8. Plug-in workflow diagram.

different types of attributes. As an example, a user could embed “Nastran Stress Analysis” data into a model and embed “Customer Reported Measured Stress”. With both datasets in the CAD model a user could compare them by assigning different colors to each type of attribute and viewing both on the same model. If it is difficult to see both datasets on top of each other the user could decide to just visualize the extreme value ranges on both data sets, with each attribute being a different color. A user could also decide to just view where “Customer Reported Measured Stress” matches the “Nastran Stress Analysis” or where they differ.

The overview diagram seen in Fig. 8 illustrates the general workflow of how a user would interact with the plug-in.

The workflow diagram seen in Fig. 8 illustrates the ecosystem that the user would be working in. As it can be seen from Fig. 8 the plug-in attempts to aid the user digestion of knowledge datasets by supplementing the capabilities of the CAD package. In the much larger PLM ecosystem any software package will be able to save and track the CAD model, with embedded information, just like it normally does with regular models, without embedded information.

3.4. Data validation

Several tests were run to see if the methods implemented allowed data to be embedded and accessed without data corruption. The methodology employed is one we will refer to as a round-trip. The methodology is to create a file, which is composed of a part, then save that file using the native format provided by the CAD package and see if data is changed using the basic saving method. The file with the original part is also saved using an NFF and checked when the NFF is reimported to the CAD see if data has changed. This will give us our base line of what normally happens within that CAD package. The goal is to have methodology to embed our data in which the data will not become corrupted or change in value when it is saved into the native CAD package format or when it is saved to an NFF and reimported back in. In order to check the base line of how files are handled a part was created and then that parts density, volume, area and mass were recorded. The part was then saved in the native file format of that CAD package and in a new instance of the CAD package the saved part was loaded in to see if loading a natively made part suffers any data loss or errors. It was determined that a part saved into the CAD packages native format and retrieved by the same CAD package does not suffer any data errors, all four measured pieces of information remained constant. However when that same part is saved out to a neutral file format, such as STEP, and then imported into the same CAD package errors in the part data are detected. The density of the part remained the same as the originally created part yet the volume, area and mass of the part began to be altered by that CAD package. In the tests, the errors remained under 1% difference from the originally created part and the loaded NFF part.

After having the base line of what we could expect we took the same created part and embedded data into the model then the same round-trip tests are run. When importing in a saved file which has embedded data in the native CAD package format there was no change in any of the measured fields as well as there was no change in the data that was embedded. The embedded data and the location it was placed in remained unchanged. Then the part with the embedded data was exported to the NFF format and the NFF was imported back into a new instance of that CAD package. While we saw the same error appear in the volume, area and mass of the part, such as with our base line round-trip tests. However, the embedded data did not suffer any error. The location and data that was embedded was untouched and remained identical to what was originally embedded. This NFF with embedded data was then imported into a different CAD package and what was seen is that the

embedded data was also not altered in the other CAD package.

3.5. Plug-in applications

The software developed is capable of storing data in the computer-aided design model, and searching through this data. This is the primary use of the tool developed, and its application in this realm is significant. This plug-in is currently best suited for structured data sets that relate to locations in the computer aided design model. This section will run through potential applications of the tool, evaluate its relationship to the digital twin and one use case that is currently being conducted.

The most significant application of the tool is for the storage of critical behavioral information. Critical behavioral information is information that will directly affect the effectiveness of the product. The critical behavior information dataset should be a structure dataset, because the tool is incapable of storing unstructured datasets. An example use would be after an analysis phase the analysis information, which is deemed critical to product behavior, can be structured and stored within the computer-aided design model. Now the enhanced computer-aided design tool acts as a simpler interface for the query of critical behavioral information. This may provide insight into potential issues that may have been overlooked prior, or simply allow for easy query and retrieval of behavioral information. Using the computer-aided design model as an index for information that has a spatial component provides significant value to the consumers of the information, and connects information of disparate roles through a common interface.

A second opportunity for the use of the computer-aided design model is through machines. Machines need different forms of communication then humans. Typically visual formats are a less effective method for communication to machines, and although the data is stored in the computer-aided design model as a visual point, it is also in the file structure. What this means is that a machine is capable of reading in the information that is stored within the computer-aided design model. An example of potential application for this is a machining simulation communicates through the computer-aided design model the predicted residual stress throughout a product in process. The machine can now make a decision for its operation in response to the probable residual stress within that product. This type of automated machine decision making can provide significant improvements to the product lifecycle.

The machine example, and the storage and query of critical information are examples of potential applications

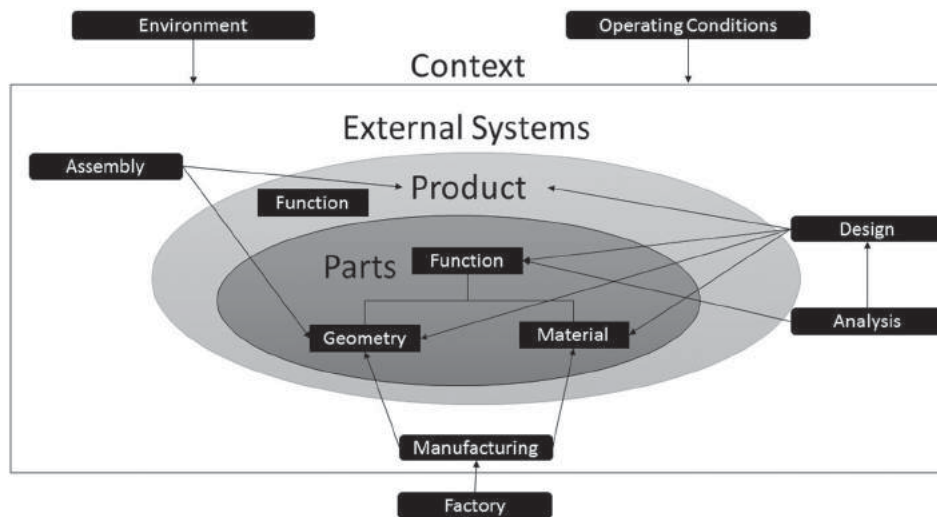


Figure 9. Model of a product as part of the product lifecycle.

for this software. They are not the sole possibilities, but are meant to communicate that this type of application could be of significant value. This software opens new doors for the integration of data and models throughout the product lifecycle, and lays a foundation for a digital twin. Fig. 9 is a representation of the complexity of the product lifecycle environment. There are a significant number of relationships through an enterprise which affect the final product. In order for the digital twin to be successful there is a need to identify the product lifecycle elements which need associations or integration. Ultimately all aspects of a product may indeed be unified, but in the short term it will be more attainable to create integrations based on the value.

Currently, this tool is being applied to a recently completed Foundational Engineering Problem which tackled the manufacture of Ni-based super alloy turbine disks. The tool will be used to visualize the bulk residual stress as a result of the forging.

3.6. Summary

This research extended the model-based definition to incorporate behavior and product characteristics directly into the computer-aided design model. The objective was to demonstrate that behavioral information could be incorporated into model-based definition through the computer-aided design model. The implementation section outlines the solution that was created to extend the MBD framework, and demonstrates that behavioral information can augment any CAD package. The background section extends the theory behind MBD in that shape alone should not constitute the definition of a product rather that the product is a multifaceted concept

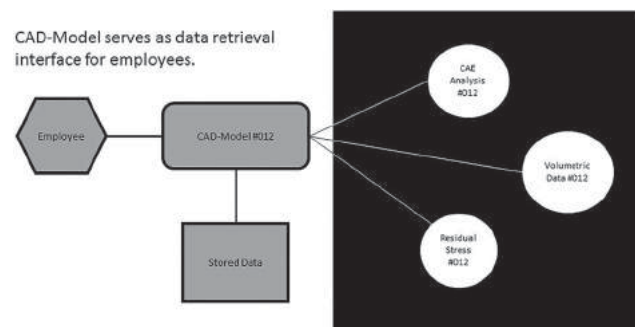


Figure 10. CAD model acts as an index for information.

composed of a many domains of knowledge that should be brought together to work in tandem. The digital twin should not simply be a replica of the physical object, but also augment the definition of the physical product.

The tool developed enables the use of the computer-aided design model, as an index for information. This reflects the integration of multiple models for the digital twin, by providing ease of access to information pertinent to the part in focus. Furthermore, the part itself enables data analytics via its geometry. Through visualizations which set normally distinct datasets against the geometry, and each other new insight is possible. This tool enables an aspect of the digital twin, by integrating information into a more holistic repository. A graphic which describe how the enhanced model enables the digital twin can be seen in Fig. 10.

4. Conclusion

The objective of this research was to extend the 3D-CAD model through behavioral information. The solution

developed achieved this, and provided insight into the benefits of such a solution. Through interaction with experts, the researchers were able to identify the potential value of an extended CAD model. The value was being able to search between and among datasets, visualization of the data sets, and downstream human or machine consumption of extended information. An extended 3D-CAD model will continue to be developed, and this research represents foundational work in the utilization of behavior in the CAD model to create a digital twin.

ORCID

Alexander McDermott Miller  <http://orcid.org/0000-0002-7675-0057>

Ramon Alvarez  <http://orcid.org/0000-0002-7838-7300>

Nathan Hartman  <http://orcid.org/0000-0002-1300-0240>

References

- [1] Bouikni, N.; Rivest, L.; Desrochers, A.: A multiple views management system for concurrent engineering and PLM, *Concurrent Engineering*, 16(1), 2008, 61–72.
- [2] Camba, J.; Contero, M.; Johnson, M.; Company, P.: Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability, *Computer-Aided Design*, 57, 2014, 61–73.
- [3] Demoly, F.; Monticcolo, D.; Eynard, B.; Rivest, L.; Gomes, S.: Multiple viewpoint modelling framework enabling integrated product–process design, *International Journal on Interactive Design and Manufacturing*, 4(4), 2010, 269–280.
- [4] Gero J.-S.; Tham K.-W.; Lee H.-S.: Behavior—a Link between Function and Structure in Design, *Ifip Transactions B-Applications in Technology*, 4, 1992, 193–220.
- [5] Glaessgen, E.-H.; Stargel, D.: The Digital Twin paradigm for future NASA and US Air Force vehicles, 53rd Structural Dynamics Material Conference Special Session, Digital Twin, Honolulu, HI, US, 2012.
- [6] Han, J.; Zhang, F.-P.; Gao, B.; Zhang, J.; He, Y.: Unified Product Information Modeling Using Model-Based Definition Technology, In *Advanced Materials Research*, 591, 2012, 837–840. Trans Tech Publications.
- [7] Hedberg Jr, T.-D.; Hartman, N.-W.; Rosche, P.; Fischer, K.: Identified research directions for using manufacturing knowledge earlier in the product life cycle, *International journal of production research*, 55(3), 2017, 819–827.
- [8] Herzog, G.; Kröner, A.: Towards an integrated framework for semantic product memories, In *SemProM*, 39–55, 2013, Springer Berlin Heidelberg.
- [9] Iwasaki, Y.; Chandrasekaran, B.: Design verification through function-and behavior-oriented representations, *Artificial Intelligence in Design'92*, Springer Netherlands, 1992, 597–616.
- [10] Ma, Y.-S.; Chen, G.; Thimm, G.: Paradigm shift: unified and associative feature-based concurrent and collaborative engineering, *Journal of Intelligent Manufacturing*, 19(6), 2008, 625–641. <https://doi.org/10.1007/s10845-008-0128-y>
- [11] Mocko G.; Malak R.; Paredis C.; Peak R.: A knowledge repository for behavioral models in engineering design. ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Jan. 1, 2004, 943–952. <https://doi.org/10.1115/detc2004-57746>
- [12] Ouamer-Ali, M. -I.; Belkadi, F.; Laroche, F.; Remy, S.; Bernard, A.: Facet-based approach for the management of information multi points-of-view in product modeling, *Computer-Aided Design and Applications*, 14(5), 2017, 582–594.
- [13] Oxman, N.; Jesse L.-R.: Material-based Design Computation an Inquiry into Digital Simulation of Physical Material Properties as Design Generators, *International journal of architectural computing*, 5(1), 2007, 25–44. <https://doi.org/10.1260/147807707780912985>
- [14] Quintana, V.; Rivest, L.; Pellerin, R.; Venne, F.; Kheddouci, F.: Will Model-based Definition Replace Engineering Drawings throughout the Product Lifecycle? A Global Perspective from Aerospace Industry, *Computers in Industry*, 61(5), 2010, 497–508. <https://doi.org/10.1016/j.compind.2010.01.005>
- [15] Stephan, P.; Eich, M.; Neidig, J.; Rosjat, M.; Hengst, R.: Applying digital product memories in industrial production, In *SemProM*, 283–304, 2013, Springer Berlin Heidelberg.
- [16] Tuegel, E.: The airframe digital twin: Some challenges to realization, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 2012. <https://doi.org/10.2514/6.2012-1812>
- [17] Umeda, Y.; Ishii, M.; Yoshioka, M.; Shimomura, Y.; Tomiyama, T.: Supporting conceptual design based on the function-behavior-state modeler, *Artificial Intelligence for Engineering, Design Analysis and Manufacturing*, 10(4), 1991, 275–288. <https://doi.org/10.1017/S0890060400001621>
- [18] US Navy Model Based Definition Initiative Identifies Significant Savings, <http://www.iti-global.com/us-navy-model-based-definition-initiative-identifiessignificant-savings>, ITI.
- [19] Utpal, R.; Bharadwaj, B.: Design with part behaviors: behavior model, representation and applications, *Computer-Aided Design*, 34(9), 2002, 613–636.