

# On the Integration of Model-Based Feature Information in Product Lifecycle Management Systems

Jorge D. Camba<sup>a</sup>, Manuel Contero<sup>b</sup>, Pedro Company<sup>c</sup>, David Pérez<sup>b</sup>

<sup>a</sup>*University of Houston, Houston, TX*

<sup>b</sup>*Universitat Politècnica de València, Valencia, Spain*

<sup>c</sup>*Universitat Jaume I, Castellón, Spain*

## Abstract

As CAD models continue to become more critical information sources in the product's lifecycle, it is necessary to develop efficient mechanisms to store, retrieve, and manage larger volumes of increasingly complex data. Because of their unique characteristics, 3D annotations can be used to embed design and manufacturing information directly into a CAD model, which makes models effective vehicles to describe aspects of the geometry or provide additional information that can be connected to a particular geometric element. However, access to this information is often limited, difficult, and even unavailable to external applications. As model complexity and volume of information continue to increase, new and more powerful methods to interrogate these annotations are needed.

In this paper, we demonstrate how 3D annotations can be effectively structured and integrated into a Product Lifecycle Management (PLM) system to provide a cohesive view of product-related information in a design environment. We present a strategy to organize and manage annotation information which is stored internally in a CAD model, and make it fully available through the PLM. Our method involves a dual representation of 3D annotations with enhanced data structures that provides shared and easy access to the information. We describe the architecture of a system which includes a software component for the CAD environment and a module that integrates with the PLM server. We validate our approach through a software prototype that uses a parametric modeling application and two commercial PLM packages with distinct data models.

*Keywords:* 3D annotations, PLM, PDM, design intent, model-based engineering.

## 1. Introduction

Model-Based Enterprise is an emerging development paradigm where 3D models and Technical Data Packages are used as the main sources of documentation and communication throughout the lifecycle of a product. In a highly globalized world where technology evolves rapidly and digital information is vital, many organizations are moving away from the dependence on technical drawings to a model-based approach to design and manufacture their products. This move represents an opportunity to work more efficiently, increase performance, and ultimately remain competitive (Frechette, 2011).

In today's engineering and design environments, digital tools are used extensively to define all aspects of a product's lifecycle. For example, Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) tools are used to create, modify, analyze, optimize, and often fabricate a product. Other tools are used to create technical documents, manage information, resources, and services. By relying on digital technology, organizations can create information networks that can be used, shared, and leveraged by all stakeholders (Regli et al., 2000; Park, Fujimoto & Hong, 2012). In this scenario, 3D models naturally become richer and more comprehensive data sources as they carry increasing amounts of design data and take the center stage of the product's lifecycle. For many design and manufacturing organizations that produce and consume large volumes of CAD data, automated and efficient solutions for managing these data are a must. These solutions often materialize in sophisticated file repositories and Product Lifecycle Management (PLM) tools, where data and processes are centralized, providing an efficient software infrastructure to coordinate product development activities.

The success of the MBE paradigm involves leveraging the benefits of annotated CAD models. When combined with PLM systems, these models facilitate the distributed development that is possible in the MBE setting (Caldwell & Mocko, 2008; Regli et al., 2000). For example, when a model is documented, both humans and computers can easily process and reuse the information, which eliminates errors, minimizes labor, and ultimately reduces the cost of supporting and maintaining the product throughout its life (Alducin-Quintero et al., 2012; Drury et al., 2013). However, the heterogeneous nature of the annotations as well as managing the links to the specific aspects of the model, make PLM integration a challenging task.

Modern CAD tools are beginning to embrace the MBE paradigm by implementing functionality to support model-based standards. This functionality is generally provided by Product and Manufacturing Information (PMI) modules. Because of the emerging nature of MBE and the fact that its use has not yet been widely adopted, PMI modules in many commercial packages are still incomplete and often provided as separate add-ons. PMI tools typically include symbols (such as dimensions and tolerances), and notes in textual format. Geometrical Product Specification symbols (GPS) [<https://www.iso.org/publication/PUB100296.html>] covers a subset of PMI with only manufacturing symbols, mainly linked to Geometric Dimensioning and Tolerancing (GD&T). The functionality is often limited to creating, editing, moving, and deleting annotations, although basic styling and formatting such as font size, color, and orientation can be adjusted. Advanced functionality such as annotation filtering, searching, and automatic processing, which are essential in collaborative environments, is currently missing. The limitations of current annotation tools are especially noticeable when working with heavily annotated models, and in collaborative environments.

In this paper, we lay the foundation for the development of a comprehensive information framework based on annotated CAD models for capturing, communicating, and sharing design knowledge. We present a software architecture that enhances the functionality of the standard 3D annotation mechanisms that are typically available in professional CAD packages by defining new data structures that can be connected and integrated with PLM systems. The proposed strategy combines the strengths of both internal and external representations to provide a richer and more comprehensive view of product-related information in an engineering development environment, which can significantly improve the information flows within collaborative CAD/CAM/CAE processes. Our approach demonstrates how standard annotation mechanisms can be used as the basis for a new infrastructure that supports information-rich CAD models where design information is embedded within the geometry. We validate our approach by implementing and testing a custom plug-in for a commercial CAD package with two distinct PLM systems with different architectures and data models running simultaneously. Technical and implementation details are described to ensure repeatability of our work.

## 2. Related work

### 2.1. Model-Based Enterprise (MBE)

Model-Based Enterprise (MBE) is an approach to product development that uses digital models to drive and coordinate all engineering activities throughout the entire product's lifecycle (Lubell et al., 2012). In this paradigm, 3D data are used as the main source of information and the entity from which other outputs and activities flow. CAD models become the vehicles that carry the product's documentation and enable collaboration among product teams (designers, analysts, technicians, manufacturers, procurement, etc.). Well implemented MBE environments have been shown to improve products, reduce time to market, and increase reusability (Boehm et al., 2010; Pellerin, Quintana & Rivest, 2013). Recent initiatives to promote MBE include the ongoing efforts of the US Department of Defense and the US National Institute of Standards and Technology (NIST) Engineering Laboratory, both of which have been actively involved in the development of MBE for a number of years and have increased visibility of the MBE paradigm in both public and private sectors (Lubell et al., 2012; DoD, 2013).

The MBE paradigm is founded on the concept of Model-Based Definition (MBD). A product's MBD can be defined as a dataset comprising the model's 3D geometry and annotations (Quintana et al., 2010). The annotations specify manufacturing and life cycle support data, known as Product Manufacturing Information (PMI), which may include Geometric Dimensions and Tolerances (GD&T), material and process specifications, and inspection requirements. To coordinate and communicate via a CAD model, it is necessary to not only define its geometry, but also to manage the associated notes and metadata that make the technical documentation of the product. The full technical description of a product is known as Technical Data Package (TDP) and consists of models, PMI, performance requirements, documentation, packaging information, and other details (DoD, 2013).

There are a number of benefits to migrate to an MBE environment. For example, many tasks can be automated, as digital data can be translated, processed, and shared easily as needed by users. The availability of design information across the system's lifecycle also leads to time savings, improved efficiency and data reuse, and ultimately cost savings for the organization (Silventoinen et al., 2014). Many companies have already transitioned (or are in the process of transitioning), to paperless environments (Lubell et al., 2012). Because CAD models in the MBE environment are so inherently rich in information, it is necessary to implement mechanisms that can efficiently manage, track, and control product data. Similarly, to ensure the robustness of information and the effectiveness of communication, it is important to guarantee consistency in the creation and presentation of the digital CAD model. Maintaining the integrity of product information is the responsibility of all users and anyone who may add or change the model during its creation and revision. In this context, standards are critical as they

dictate the necessary guidelines for efficient implementations and performance.

Two standards are relevant to the PMI approach based on annotated 3D models: ASME Y14.41 (Digital Product Definition Data Practices) (ASME, 2012) and ISO 16792-2015 (ISO, 2015). Both standards distinguish between the *design model* and the *annotated model* (or simply *model*). The design model is the “portion of the data set that contains model geometry and supplemental geometry.” The model is “a combination of design model, annotations, and attributes that describe a product.” In other words, design models become models when they are enriched by *attributes*, which are specifications that define certain properties (i.e. a type of metadata attached to the model). *Annotations* are instantiation of attributes (annotations display information of attributes).

Different types of annotations are defined: dimensions, tolerances, note, text, or symbols. The rules to display annotations are based on *annotation planes*, which are conceptual planes that either perpendicularly intersect or are coincident with one of the surfaces of the annotated feature. There are significant differences between symbols (in general, and also in the particular cases of dimensions and tolerances) and text (both notes and free text). Symbols are strictly defined—and thus are compact and strongly constrained—whereas text is free and more extensive than symbols, which makes it more difficult to process and automate. In addition, both ASME and ISO standards describe the expected behavior of the digital model when an attribute is queried: the corresponding annotation and its related information should be displayed. This action applies not only to the original CAD system, but also to any additional plug-ins.

Other standards such as the “SASIG 3D Annotated Model Standard” (SASIG, 2008) for the automotive industry, build on ASME and ISO guidelines to define how to work with free textual annotations. According to SASIG, the use of groups, layers, and links to views or sections of the geometry are recommended to make the model readable, as annotations in a complex model may quickly make visualization difficult (SASIG, 2008). In this paper, the previous standards are used to guide the development of the annotation structures that interact directly with the CAD environment and the user.

## 2.2 Management of 3D annotations

Various approaches and software systems have been suggested to manage 3D model annotation information. For example, Davies (2008) developed a hybrid annotation framework for both semantic and procedural annotations that allows multiple viewpoint annotations. Another semantic approach based on viewpoints was proposed by Aubry et al. (2007) and Thouvenin et al. (2005). Their structured systems maintain annotations into three viewpoints: design concept (materials, scientific and technical domains, etc.), geometrical description, and methods, and uses an ontology-based knowledge management system to process annotation information. The system “Funnotation” (Kitamura et al., 2006) is also relevant, as it was developed to aid CAM. Its complete semantic annotation model relies on processing text documents and contains four elements: the function of the device, how the function is achieved, the functional decomposition structure, and alternative solutions to achieve the functions. When the system is filled with sufficient annotation entries, it can automatically suggest suitable parts based on a given functional design specification.

The solution proposed by Bilasco et al. (2006) stores a set of semantic profiles of 3D scenes. Users can send requests to the annotation repository and retrieve the specific information pointing to corresponding 3D fragments. An integration component translates all fragments and reassembles them into a new 3D model according to the topology defined by the request.

For the purposes of this paper, the most relevant points on managing 3D annotations are summarized below. The summary is used as the basis to justify our decision to use a hybrid representation approach in our solution:

- 3D Annotations can be classified as “in-line” (internal) or “stand-off” (external), based on how the information is stored. Internal annotations are efficient in terms of information processing and manipulation and are also easier to implement. In fact, many CAD systems natively support this type of structures. However, internal annotations are difficult to share in collaborative environments and creating multiple sets of markup schemes for a single model is problematic (Ding & Liu, 2010).
- External annotations are more flexible, as they support traceability, progressive updates (scalability), information reorganization for different purposes, and easy distribution in collaborative environments. Some disadvantages include the problem of persistent references and increased efforts in terms of file maintenance (Ding & Liu, 2010; Mun & Han, 2005).
- In an extensive literature review paper, authors Chandrasegaran et al. (2013), recommend a mixed representation of annotations by stating that the combination of internal and external representations facilitates integration and

reduces workload so users can focus on the design work, instead of learning a new software tool, and thus, avoid the knowledge-acquisition bottleneck.

- In terms of usability, the “stand-off” functionalities are especially relevant to manage the visual clutter created when the number or the arrangement of the annotations in a 3D model causes confusion in the user to the point where they negatively affect performance (Camba, Contero, & Johnson, 2014). This is particularly relevant in collaborative environments where multiple users may need access to different information within the same annotated model.

### *2.3 Product Lifecycle Management (PLM) Systems*

PLM is a broad concept that refers to the management of all product-related information and processes throughout its entire lifecycle (Ameri & Dutta, 2005). The term is typically connected to other business technologies such as Supply Chain Management (SCM), Customer Relationship Management (CRM), and Enterprise Resource Planning (ERP), and form the foundation of a company's information technology infrastructure. According to Sharma (2005), collaboration at the highest level involves connecting people, processes, and information. This connection requires an underlying PLM framework that facilitates communication, as it is through communication that teams make sense of the information needed for their work, their shared sense of who knows what, and the shared meaning of the ideas necessary to knowledge work (Yuan et al. 2010). However, despite the variety of PLM software available on the market, the lack of interconnectivity with other enterprise information systems is still a common challenge in professional environments (Soto-Acosta, Placer-Maruri, & Perez-Gonzalez, 2016).

A major building block of a PLM system is the Product Data Management (PDM) component, which tracks and manages the information generated during the product lifecycle with the purpose of improving productivity and reusability of product data. This information includes digital product models, assemblies, analysis results, engineering change management information, and component libraries, among others. The PDM serves as a central repository for process and product history, and promotes integration, collaboration, and data exchange among all stakeholders by providing better control of the information, engineering activities, design changes, and product configurations.

The structure of a PDM can be described at three levels. The first level is the information repository where information is stored. The information repository controls data access, storage and recall, information security and integrity, concurrent use of data, and archival and recovery. It is also responsible for the traceability of all actions taken on data. The second level provides user access, permissions, and control policies to the information stored in the repository. The third level serves as an interface for users and programs to access the data and is responsible for user queries, user input, and report generation. It also provides interfaces for CAD programs to interact with the PDM. It is in a PDM context, specifically at the first level (information repository), where our system is relevant. The proposed addition provides a mechanism to track and manage annotated 3D models, thus making the information accessible to external processes, users, services, and applications.

## **3. System Architecture and Data Structures**

Our proposed strategy implements a software module that focuses on actions that cannot be performed by PMI tools alone, such as synchronization, grouping, filtering, searching, efficient visualization of information, and lifecycle management (upload, download, locks, unlocks, states, revisions, etc.) The two major elements of the system include an embedded annotation management component and a unit responsible for PLM interaction.

The system overcomes the limitations of standard PMI modules by using an external repository to store enhanced information about the internal annotations. To this end, two structures are used to manage annotation information: an internal structure created by the CAD package and stored within the model; and an external representation, stored separately, that contains the enhanced data. The two structures are synchronized automatically by a background process in a manner that is transparent to the user. “In-line” techniques are maintained so users can annotate the model with the standard PMI tools that are already available in the CAD environment. The “stand-off” characteristics of our implementation facilitate the visualization, search, and filtering of information. In addition, by making annotation data available to external applications, the information can be processed by other applications to automatically generate design knowledge.

To ensure bi-directionality, the two representations are synchronized by a background process that is triggered every time an annotation is created, modified, deleted, and when the model is saved or uploaded to the PLM server. Additionally, if inconsistencies occur, the module checks data integrity and maintains the most up-to-date representation. If an inconsistency cannot be resolved, the system prompts the user for action, i.e., to discard model or annotation changes. In our system, the

two representations are connected by internal identifiers. For every annotation, a unique “id” is created by the PMI module and stored both inside the CAD model and externally. The external data structure contains a set of elements that describe the internal annotation and extend its capabilities. These elements are:

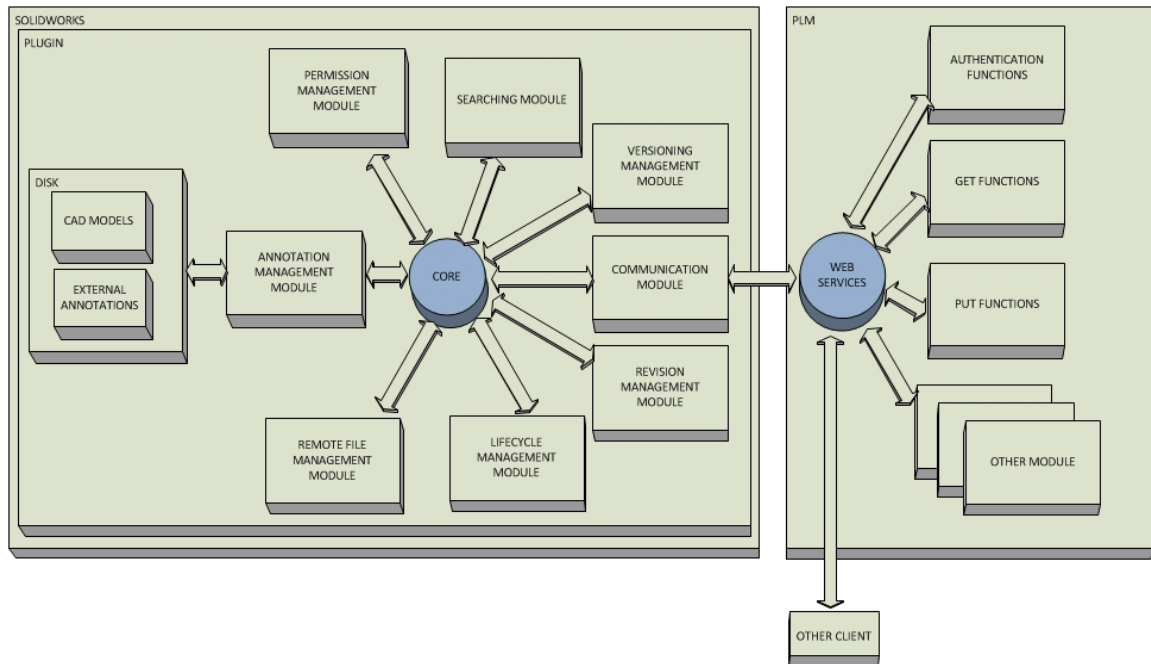
- *feature*, the specific geometric element in the annotated CAD model.
- *text*, the annotation content in textual form.
- *creator*, the name of the creator of the annotation.
- *date* of the last modification of the annotation.
- *internalId* and *persistentId*, a unique annotation identifier.
- *posX*, *posY*, and *posZ*, represent the point in 3D space where the annotation is located.
- *website*, a hyperlink to an external website or file connected to the annotation.
- *deleted*, indicates whether the annotation was removed from the model.
- *likes*, is a base64 encoded text separated by a special character that contains a list of the users who considered the annotation useful.
- *dislikes*, similar to the previous field, but representing users who tagged the annotation as not useful or interesting.
- *annotationhistory*, records annotation changes. It saves the original text before modification (*text*), the date when the text was modified (*modificationDate*), and the user information (*creator*)

The above data are stored in an external XML file. Every CAD model has an associated XML file, where each annotation in the model is saved as a node. A sample file is shown in Fig. 1.

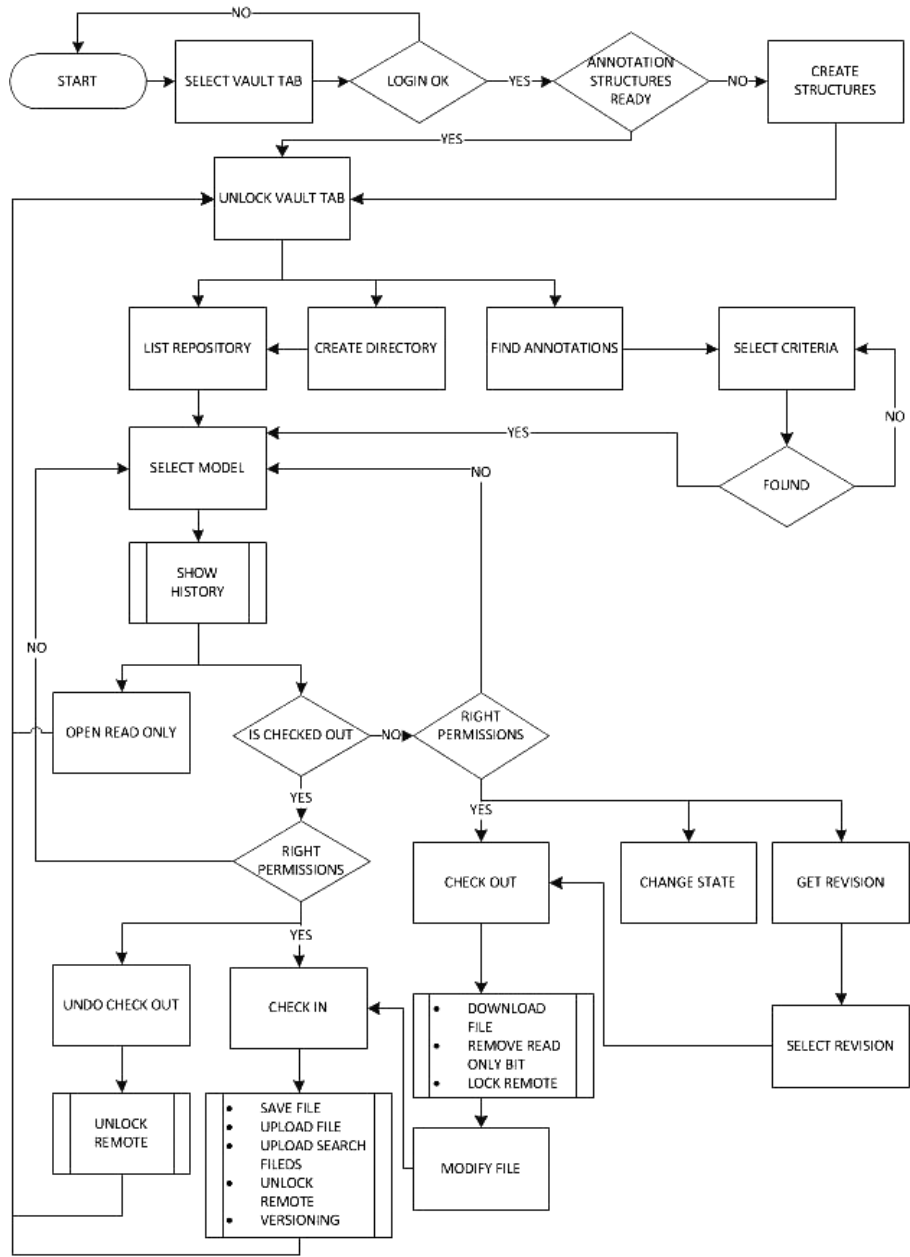
```
<?xml version="1.0" encoding="UTF-8"?>
- <annotationsdef>
  <write_date>14/12/2015 8:02:19</write_date>
  - <annotation>
    <feature>Base-Revolve</feature>
    <text>bevel v2</text>
    <creator>Dapelo</creator>
    <date>09/12/2015 16:16:58</date>
    <internalId>Elemento de detalle4</internalId>
    <persistentId>0880270000000030000000002552542550000000000000025525500100001100
    <posX>-0,0358368457431386</posX>
    <posY>0,0069856749169721</posY>
    <posZ>0,01417</posZ>
    <website>info.pdf</website>
    <deleted>False</deleted>
    <likes>YWRtaW4yI0AI12FkbWluMiNAJSNhZG1pbjIjQCUjYWRTaW4yI0AI12FkbWluMw==</likes>
    <dislikes>YWRtaW4=</dislikes>
  - <annotationhistory>
    <text>surface</text>
    <modificationDate>14/12/2015 13:33:51</modificationDate>
    <creator>Dapelo</creator>
  </annotationhistory>
  - <annotationhistory>
    <text>bevel</text>
    <modificationDate>14/12/2016 14:52:09</modificationDate>
    <creator>Dapelo</creator>
  </annotationhistory>
</annotation>
</annotationsdef>
```

Fig. 1. Sample XML file

To integrate our system within a PLM environment, both the CAD model and its associated XML file are stored in the PLM server. The system automatically connects to the PLM and efficiently stores, retrieves, and synchronizes all annotations. Additionally, because of the nature of the XML file, the proposed system is designed to work offline by only synchronizing changes when a PLM connection is available. The functional blocks of the proposed architecture are shown in Fig. 2. The flow chart of the operations performed against a PLM is shown in Fig. 3.



**Fig. 2.** Functional blocks of the proposed system



**Fig. 3.** Flow chart of PLM operations

The proposed design consists of a main core unit and a set of modules to control models, annotations, and PLM operations. The core unit interconnects and synchronizes all modules and communicates with the PLM via API calls against PLM web services. In this scheme, the PLM can be considered as a service provider to clients. These services include functions to “put” data (check in, set lifecycle state, set revision, lock/unlock data, etc.), “get” data (check out, get revision, get version, etc.), and authenticate. Each module manages a specific type of data, builds the queries that need to be performed by the communications module, and listens to the responses from the PLM. The core unit is responsible for collecting data from the communications module and sending it to the corresponding module, and vice versa.

The annotation management module is a key component of the system. It manages CAD models, annotations, and disk operations. The remaining modules perform PLM operations. The functionality of each module is described in Table 1.

**Table 1.** Functionality per module

Module	Function
Permission management	<ul style="list-style-type: none"> <li>Collect authentication data.</li> <li>Establish accessibility and permissions based on user profiles.</li> </ul>
Remote file management	<ul style="list-style-type: none"> <li>Check-in and check-out operations.</li> <li>Transfer CAD models and external annotations from the local machine to the PLM server, and vice versa.</li> </ul>
Versioning	<ul style="list-style-type: none"> <li>Manage document versions (i.e., it retrieves information about a version, sets a new version, and retrieves a certain version of a document).</li> </ul>
Revision management	<ul style="list-style-type: none"> <li>Control document revisions.</li> <li>Return documents to earlier revisions.</li> </ul>
Lifecycle management	<ul style="list-style-type: none"> <li>Retrieve and update information about the document lifecycle state.</li> </ul>
Search	<ul style="list-style-type: none"> <li>Allow the user to search files in the repository according to certain criteria.</li> <li>Check for the required structures to perform a search. If they do not exist, it creates them.</li> </ul>
Annotation management	<ul style="list-style-type: none"> <li>Manage CAD models, annotations, and disk operations.</li> </ul>
Communication	<ul style="list-style-type: none"> <li>PLM interaction</li> </ul>

## 4. Implementation

### 4.1 Annotation management

The annotation management module is the component in our system that is fully integrated with the local CAD environment (SolidWorks, in our case) and directly interacts with the internal annotations in the CAD model. Visual Basic was used for implementation purposes, as it is natively supported by both SolidWorks and the systems used for PLM integration. SolidWorks was selected as a testing system for convenience and availability reasons. The increasing popularity of the MBE paradigm has pushed CAD vendors to develop tools that support basic annotation functionality. Today most major mechanical CAD packages (e.g. Autodesk Inventor, Siemens NX, Siemens Solid Edge, PTC Creo, Dassault Systems CATIA, Dassault Systems Solidworks) support model-based definition standards such as ASME Y14.41-2012 and ISO 16792:2015. Extensive work on the PMI capabilities in CAD systems (Cheney & Fischer, 2015) has also been carried out over the past few years by the National Institute of Standards and Technology (<https://go.usa.gov/mGVm>). In this regard, the system architecture and implementation details described in this paper could be applied to any PMI module, regardless of the CAD environment.

Our system provides a fully functional web browser that is built in the CAD interface, so all product-related information is available to the user from an already familiar interface: the design environment. The annotations are tabulated to facilitate direct queries based on filtering and search tools. For each annotation, related information such as feature, text annotation, author, modification date, related link, and internal id is saved to an XML file, as described earlier in the System Architecture section. Custom filtering and search functionality can be activated via the *filter selector*. Search criteria include text (the annotation text), feature (specific part of the model where the annotation is connected to), creator (author of the annotation), and date (last modification date of the annotation text). The option to *Filter in Model* is provided to reduce the amount of annotations in the viewport based on the specified criteria. CAD users have reported increased performance and time savings when filtering tools are used (Camba et al., 2014).

Finally, all annotation changes in a 3D model are recorded for tracking and archival purposes. When an annotation is altered (i.e., deleted, moved, or its content is modified), a new entry is created in the XML file of the corresponding annotation with the original information, author and date.

### 4.2. PLM integration

Our previous exploration into the integration of annotation information into a PLM system involved the use of relational databases (Camba & Contero, 2014). However, the lack of standardization and the proprietary nature of many PLM systems, made the implementation difficult, since managing PLM databases directly as well as locking and unlocking operations are



complex tasks. More importantly, annotations cannot be processed directly by the PLM server. To overcome these difficulties, and given the variety of currently available PLM systems, a decision was made to connect multiple solutions as suggested by Gunpinar & Han (2008).

As part of this work, a proof of concept of the proposed architecture was implemented. To this end, the PLM integration capabilities of our annotation manager were tested against two different systems: Autodesk Vault Professional 2016 and Aras Innovator v11. The former was selected for availability reasons and because of its extensive community support. The latter is an open source PLM system that is gaining considerable attention in industry. In our implementation, both systems are running in parallel. From a user standpoint, all PLM (Autodesk Vault or Aras Innovator) functionality is transparent.

Both Autodesk Vault and Aras Innovator use a Microsoft’s Internet Information Server (IIS) and SQL database server infrastructure for storage, fast indexing, and search of data vaults. There are three main components: a web server, a database, and a file storage system, as shown in Fig. 4. The web server hosts a set of web services to manage communication between the clients and the PLM. The database server stores file information (e.g., file status, file edit history, and property data) in the file storage system. The file storage system is a hierarchy of folders where the server stores the physical copies of the files managed by the PLM. The main differences between the two selected PLM systems are shown in Table 2. Autodesk Vault allows both local and active directory accounts, as well as custom and web-based access. The strength of Aras Innovator lies in its innovative data organization around items and relationships.

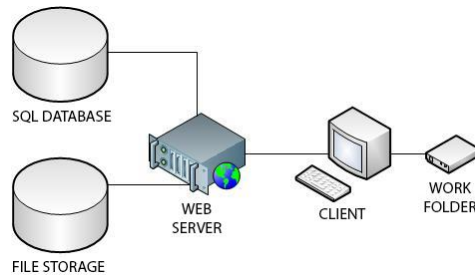


Fig. 4. PLM architecture

Table 2. Comparison of selected PLM systems

Key Features	Autodesk Vault	Aras Innovator
License	Commercial	Free
User accounts	Local and Active directory	Local
Native client	Custom client and web browser	Web browser
Project management tool	Not included	Included
Technology	Model-driven	Model-based
Data organization	Directories	Items and relationships

A variety of clients can be used to interact with Autodesk Vault PLM server. For example, a standalone application (*Autodesk Vault Client*) as well as plug-ins for various CAD packages such as *Autodesk Vault Addin for SolidWorks*, are available. Limited functionality is also provided by web clients or “thin clients,” which restrict the PLM operations that the user is allowed to perform. Alternatively, the Aras application framework uses a run-time model-based technology to define and manage business objects (Aras, 2016). The client side is web-based, but an API to develop custom applications is provided. The SolidWorks connector for Aras can be used to interface with the CAD package. Aras *CAD Documents* can be processed for SolidWorks drawings, assemblies, or parts represented in the CAD hierarchy (SW, 2013).

When a model is added to the external repository, its corresponding external annotations are uploaded automatically. The Aras API is based on *Items*, which may have relationships to other items. In our software, models are stored as *CAD Document* Items with two relationships to *File* Items: one containing the model, and the other containing the external annotations (i.e., the XML file). The structure of a *CAD Document* Item is visible from the web client and its name is an indicator of its content. Finally, *CAD Document* Items are contained within a *Directory* Item, which organizes the models hierarchically. In the case of Autodesk Vault, the *Attachment* structure is used to store and link a file (i.e., the XML file) to another source file.

Although both Aras and Autodesk versions of our system store the external annotation files, specific annotation information is not directly accessible within the repository. Our method solves this problem by avoiding a direct manipulation of the PLM

database. User-defined properties such as author, comments, and description in files from Autodesk Vault, as well as additional custom properties are used to maintain external annotation information. This solution is easy to implement and allows external applications to process the data.

In the Aras implementation, a different strategy was used to provide easier, faster, and more flexible searches over the annotations. The strategy involves creating a new *ItemType* (defined as *AnnotationItemType*), which represents an annotation and contains all possible search fields and a reference to the *CAD Document*. This *AnnotationItemType* can be seen as a new table in the PLM database (see Fig. 5).

The search functionality allows users to locate files in the repository based on user-defined criteria. These criteria include file name, comments, annotation text, modification date, annotation author, the referenced model part (feature), and annotation likes and dislikes. Annotation likes and dislikes are indicators of the usefulness of a particular annotation based on user ratings. This functionality was designed as an effort to promote enterprise social networking, as described by Mäntymäki & Riemer (2016).

Hyperlinks and external files can also be linked to annotations. Hyperlinks and filenames are saved to the external annotations XML file when the model is checked in to the PLM. External files are uploaded as *File Items*. Additional relationships to the *CAD Document* Item are created.

Number	Name	CAD_Document_...	ann_text	Parent_Directory	Feature	Creator	Date	Likes	Dislikes	cad_from [...]
00DEFB6CC0444...	motor.sldprt	89DB4000DDE84A...	video2	\$/r4/	Not attached...	Dapelo	06/04/2016...	0	0	Dapelo@#\$Dapelo
56E3817DEEC94...	circlip.sldprt	AE565AFA76E144...	otra	\$/modelos2015/	Not attached...	Angelica	08/06/2016...	0	0	Angelica@#\$Angel
5EE8DDCF9D154...	motor.sldprt	89DB4000DDE84A...	weblink	\$/r4/	Not attached...	Dapelo	06/04/2016...	0	0	Dapelo@#\$Dapelo
8981C1912EF04...	circlip.sldprt	AE565AFA76E144...	vieja	\$/modelos2015/	Not attached...	Angelica	08/06/2016...	0	0	Angelica@#\$Angel
8F1C6DA1CDC54...	motor.sldprt	89DB4000DDE84A...	pdf2	\$/r4/	Not attached...	Dapelo	06/04/2016...	0	1	Dapelo@#\$Dapelo
ABEFFAE581304...	motor.sldprt	89DB4000DDE84A...	eje2	\$/r4/	motor shaft	Dapelo	10/05/2016...	0	0	Dapelo@#\$Dapelo
B34179F8B9B84...	motor.sldprt	89DB4000DDE84A...	pdf1	\$/r4/	Not attached...	Dapelo	06/04/2016...	5	1	Dapelo@#\$Dapelo
B631979AB1644...	motor.sldprt	89DB4000DDE84A...	nueva	\$/r4/	motor core	Dapelo	28/04/2016...	0	0	Dapelo@#\$Dapelo
BCB65AE599204...	motor.sldprt	89DB4000DDE84A...	chaflán	\$/r4/	motor core	Dapelo	10/05/2016...	0	1	Dapelo@#\$Dapelo
DB5EE1AC888C4...	Fixed Gear.SLDP...	DDA34F4D952740...	Nota no asignad...	\$/modelos2016/	Not attached...	Mcont	20/06/2016...	0	0	Mcont@#\$Mcont 2
DF2CB27AF0894...	Fixed Gear.SLDP...	DDA34F4D952740...	Verificar diámetro	\$/modelos2016/	Not attached...	Mcont	20/06/2016...	0	0	Mcont@#\$Mcont 2
EE03A10165E54...	circlip.sldprt	AE565AFA76E144...	nueva2	\$/modelos2015/	Not attached...	Angelica	08/06/2016...	0	0	Angelica@#\$Angel
FD60EB6E3B7C4...	motor.sldprt	89DB4000DDE84A...	video1	\$/r4/	Not attached...	Dapelo	06/04/2016...	0	0	Dapelo@#\$Dapelo

Fig. 5. *AnnotationItemType* data on the browser

File version control is managed slightly differently by the two implementations. In Autodesk Vault, new versions are created automatically when a model is uploaded; in Aras, versions are managed by the *CAD Document* Item. When a new file is uploaded, the system deletes the related *File Items* before the new ones are added. Previous *File Items* remain in the repository and can be retrieved by selecting the specific version in the *CAD Document*.

Both Aras and Autodesk Vault provide similar mechanisms to define and manage lifecycle states and revisions. User-defined lifecycle types are flow diagrams that represent stages within the product development cycle. In our architecture, the same lifecycle is defined for both PLMs based on the typical manufacturing lifecycle process for release control. The states include: (1) *Work in progress*, (2) *For review*, (3) *Released*, and (4) *Obsolete*. Permission to add or edit product lifecycle states is only available to PLM administrators.

Model development always begins as *Work in progress* and does not change to *For Review* until the user explicitly indicates so. The next available transitions are to *Released* or to *Obsolete*. In the first case, the model is finished and ready to use. The second case is used for situations where problems are detected or changes need to be made. The original model is saved as *Obsolete* and a copy is automatically created as a *Work in progress*. Finally, when a model is labeled as *Released*, it can be used as a template for new models. The original model can be changed to *Obsolete*, if necessary.

In terms of the revision control mechanism, both PLM systems allow the user to save relevant milestones or changes to a document and its related files. Older copies of the files are maintained by our tool, which can be retrieved at any time. In the case of Aras, some aspects can be automated. For example, the system can automatically increment the revision tag when the lifecycle state is set to *Released*.

Finally, the file transfer functionality is described. To download a model the user must *CheckOut* the file, which can only be performed if the user has the right permissions and the model was not checked out by another user. The *CheckOut* function locks the file in the repository and downloads a copy to the local work directory. When a model is *CheckedOut* other users can open it as “read only.” Similarly, the upload function, or *CheckIn*, can be performed if the user has the right permissions and the model has been previously checked out. The operation saves all changes to the file, adds user comments, uploads the file to the server, and unlocks the repository (making the file available to other users). Since the operations *Get Revision*, *Change State*, and *Revise* only work with unlocked files, an *Undo CheckOut* button is provided for situations where the administrator needs to manually unlock a file and discard changes.

## 5. User Interface

The user interface of the system is based on the WIMP paradigm (Windows, Icons, Menus, Pointer). The proposed module runs inside the CAD environment and is designed as a panel to facilitate interaction. The icons are organized around a basic set of commands that are easily accessible. The user interface is compliant with ASME Y14.41-2012, as selecting items in the annotations list automatically triggers a visualization process to display the associated item in the model.

### 5.1. Annotation manager interface

The annotation manager user interface was implemented as a plug-in for a professional CAD system (i.e., SolidWorks) and designed as an additional panel in the CAD interface. The functionality of the annotation module is organized into four tabs: annotations, history, browser, and vault.

The “annotations” tab provides a tabular view of the annotations that are available in the active CAD model as well as basic search and filtering tools (see Fig. 6). For each annotation, related information such as feature, text annotation, author, modification date, related web link, likes, dislikes, and internal id, is displayed. When a user selects a specific cell in the table, the corresponding annotation is highlighted in the CAD model. If a hyperlink or associated file is available, the user can view the content in the “browser” tab, which provides a fully functional web browser, a multimedia player for audio and video files, and a PDF viewer that is built directly into the CAD interface. As a result, users do not have to leave the CAD environment to access product-related information. The “browser” tab also contains two buttons for users to “vote” whether they consider a specific annotation useful. The system automatically stores codified usernames in the XML file of users who “voted,” in order to limit the number of times a user can vote for a particular annotation (see Fig. 7).

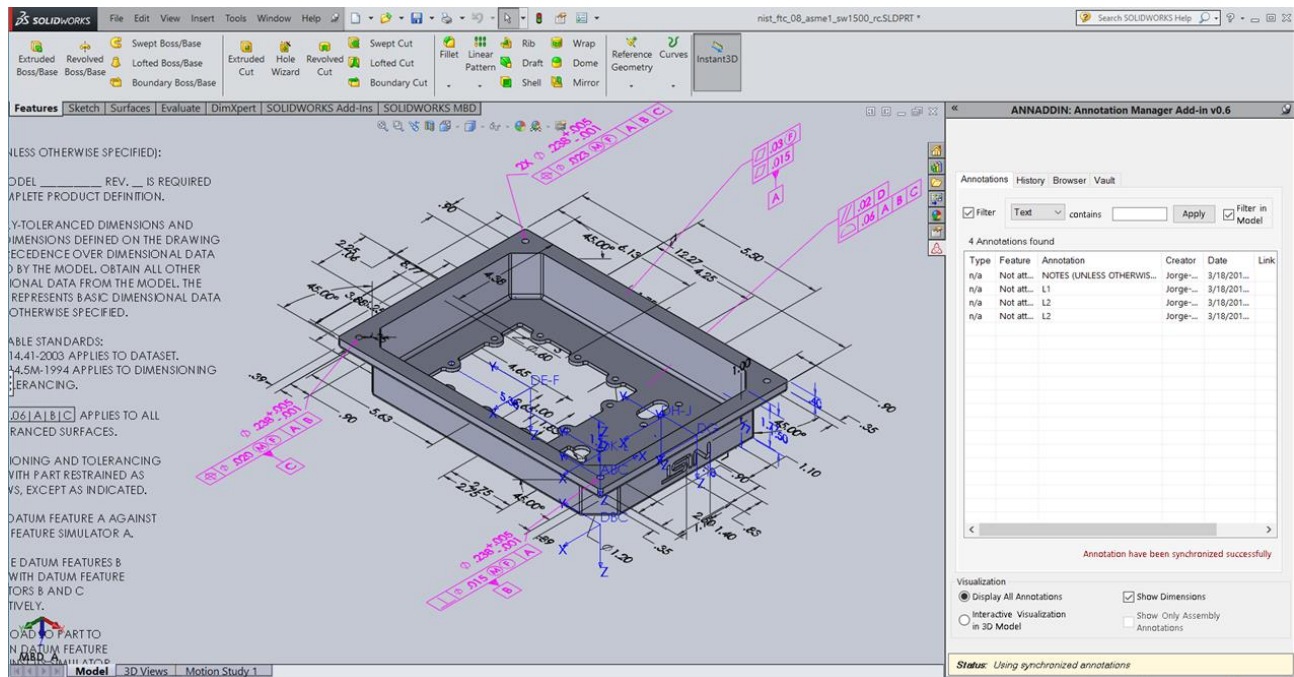
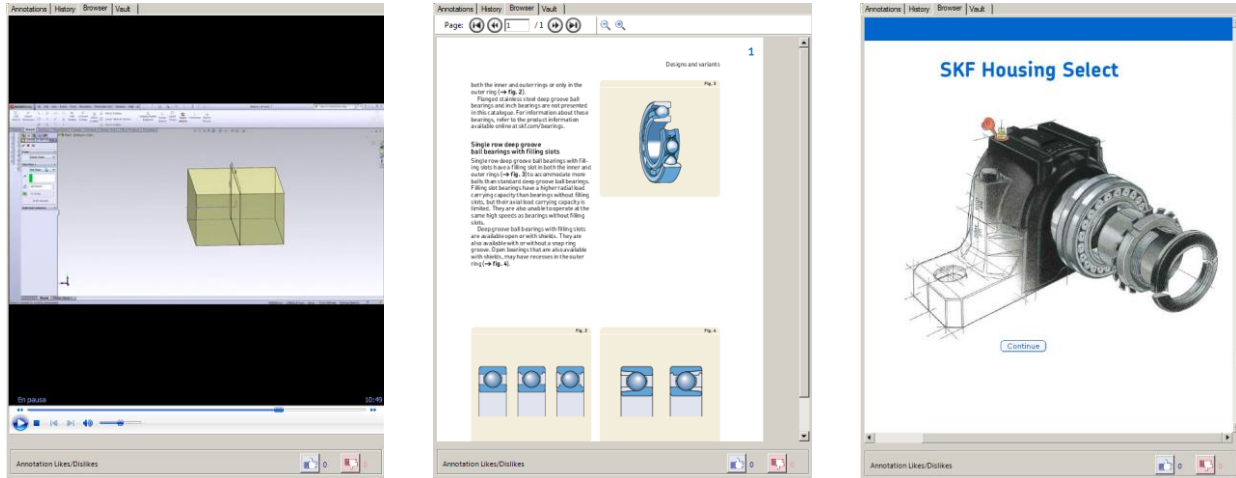


Fig. 6. CAD environment with module interface (right) and annotations tab active

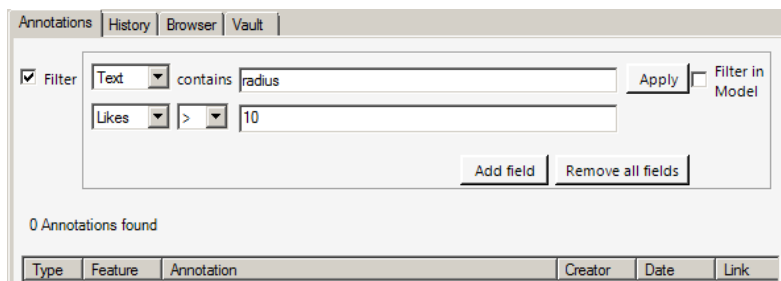
Search and filtering functionality can be activated via the “filter” selector (see Fig. 8), which toggles a text box and a pull-down menu with various search criteria, such as text (the annotation text), feature (specific part of the model where the annotation is connected to), creator (author of the annotation), date (last modification date of the annotation text), likes and dislikes. Users can combine multiple fields when searching for annotations (see Fig. 8). The option *Filter in Model* reduces the amount of annotations on the viewport based on the specified criteria, as shown in Fig. 9.



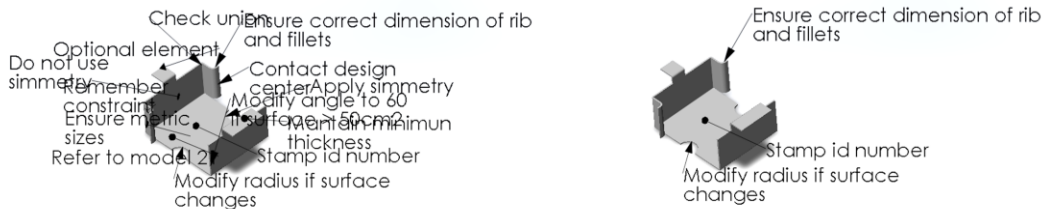
**Fig. 7.** Browser tab showing a video file (left), a pdf document (center), and a webpage (right). Likes/dislikes buttons are available at the bottom right corner of the tabs.

Additional filters are provided for grouping and various types of visualization. “Show dimensions” toggles dimensions and tolerances on and off in the model; “show only assembly annotations” toggles annotations created at the assembly level on or off (annotations created at the part level are not affected); the radio buttons “display all annotations” and “interactive visualization in 3D model” allow the user to either make all the annotations visible or view only the annotations that are currently selected in the model.

The historical record of an annotation is available in tabular form via the “History” tab from the main interface.



**Fig. 8.** Filtering tools



**Fig. 9.** Annotation visualization in model: all annotation displayed (left) and filtered annotations (right)

## 5.2. PLM interface

PLM connectivity in our system is available via the “Vault” tab, where the user must initially validate her credentials. The option to authenticate against an Active Directory is also available for Autodesk Vault. Once authenticated, users can browse the PLM repository, create files and directories remotely, and search for annotations in remote files. Repository contents are listed hierarchically as a directory structure.

Search results are displayed in tabular form where each row represents an annotation. Eight information columns are provided: file, related feature, annotation text, annotation author, remote path, modification date, annotation likes and dislikes. Results can be sorted by column. When an annotation is selected, users can open the model in “read only” mode or check out the file (download it to the local machine and lock it in the remote repository).

When a remote directory is listed and a model is selected, an additional table is displayed at the bottom of the panel, where each row represents a different version of the model, as shown in Fig. 10. The columns indicate the document version number, document revision code, document lifecycle state, author, upload date, and comments related to the document and annotation likes and dislikes.

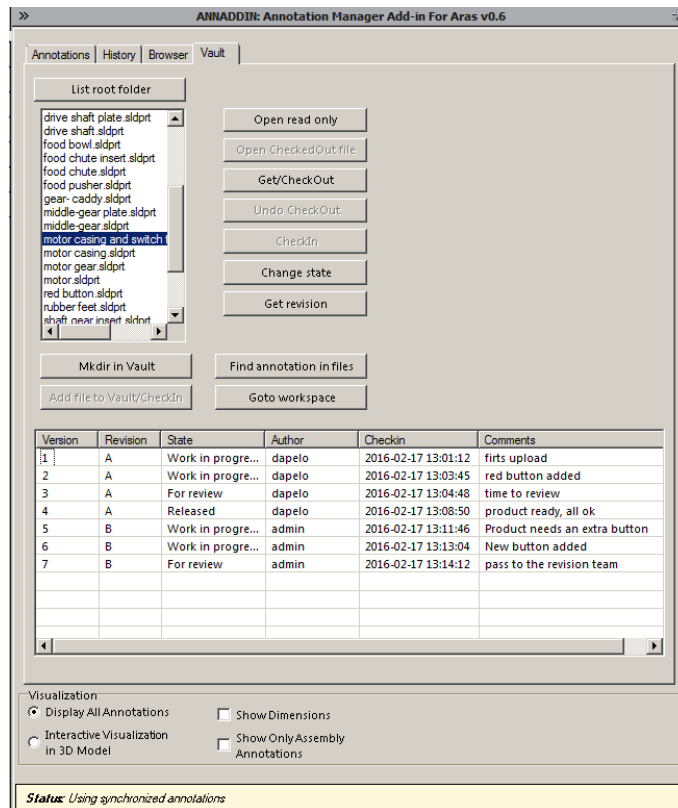


Fig. 10. Vault tab showing file version history

## 6. Conclusions and Future Work

As the Model-Based Enterprise paradigm gains popularity and 3D models continue to carry increasing amounts of design data, the need for methods and tools to support, manage, and present this information also increases. In addition, as models become more complex, there is also a need to capture and embed design information, so models can be reused easily.

In this paper, we described the architecture of a software module for a CAD system that leverages model annotations to communicate design information and seamlessly integrates into a PLM environment. The availability of annotation data through the PLM creates a new source of product-related information that is directly relevant to the CAD model and can be searched, reused, and automatically processed. Furthermore, our approach relies on annotated CAD models as carriers of information, which eliminates the need for separate tools and provides a collaborative space that is built directly into the



design environment. In this regard, our work lays the foundation for the development of a comprehensive framework based on annotated CAD models for capturing and communicating design knowledge.

To validate our strategy, we implemented a prototype for a professional CAD tool and successfully connected it to two PLM systems with different internal data models running simultaneously. The prototype was efficiently integrated in a typical production workflow and in all cases annotation information was effectively stored, retrieved, and synchronized.

The strategy described in this paper is designed to overcome the limitations of current annotation tools in collaborative environments; especially in models with large numbers of annotations. To maximize the benefits of annotation mechanisms, our approach combines the strengths of both internal and external representations via an annotation management module. To ensure bi-directionality, the external and internal representations of the annotations are synchronized by a background process.

Based on our experience and lessons learned during the development and testing of our prototype, more sophisticated grouping strategies will be explored in future versions, for example, by defining more annotation groups such as selected vs. not selected, and related vs. not related. We are interested in exploring what the best grouping strategies are, from a user interaction point of view, for annotation information in a CAD environment, as there are currently no user studies that can help us determine the most effective and useful approaches in terms of productivity. Future goals also include an extensive validation of our system in a collaborative production environment to assess the extent to which design communication can be improved by annotation-based strategies. We also plan to compare the two version revision mechanisms in the PLM: automatic vs. manual implementations.

Additionally, a web based client is currently under development to make annotations and version history information available, which can be valuable in a variety of situations. Some of the aspects that could be studied to determine what makes annotations effective in design environments include product evolution, user and team interactions overtime, and annotation quality. Novel quality metrics and techniques for improving CAD model reuse could also be inspired from this study. Finally, in terms of usability, we plan to improve the graphical aspect of the interface by including file previews, highlighting, and color schemes to facilitate navigation to study how these mechanisms impact the perception of annotation information when viewed directly on the 3D model's geometry in a CAD environment.

## Acknowledgements

This work was supported by the Spanish Ministry of Economy and Competitiveness and the FEDER Funds, through the ANNOTA project (Ref. TIN2013-46036-C3-1-R).

## References

- Alducin-Quintero, G., Rojo, A., Plata, F., Hernández, A., & Contero, M. (2012). 3D Model Annotation as a Tool for Improving Design Intent Communication: A Case Study on Its Impact in the Engineering Change Process. In *Proceeding of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 349–356).
- Ameri, F., & Dutta, D. (2005). Product lifecycle management: Closing the knowledge loops. *Computer-Aided Design and Applications*, 2(5), 577–590. <http://doi.org/10.1080/16864360.2005.10738322>
- Aras framework. (n.d.). Retrieved February 24, 2016, from <http://www.aessis.com/Services/ArasSOA>
- ASME. American Society of Mechanical Engineers. (2012). *ASME Y14.41 – 2012: Digital Product Definition Data Practices*.
- Aubry, S., Thouvenin, I., Lenne, D., & Olive, J. (2007). A knowledge model to read 3D annotations on a virtual mock-up for collaborative design. In *Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design, CSCWD* (pp. 669–674). <http://doi.org/10.1109/CSCWD.2007.4281516>
- Bilasco, I. M., Gensel, J., Villanova-Oliver, M., & Martin, H. (2006). An MPEG-7 framework enhancing the reuse of 3D models. In *Proceedings of eleventh international conference on 3D web technology* (Vol. 1, pp. 65–74). Columbia, Maryland. <http://doi.org/10.1145/1122591.1122601>
- Boehm, Barry, Bayuk J, Desmukh A, Graybill R, Lane JA, Levin A, et al. (2010). Systems 2020: Strategic Initiative.

Technical Report SERC-2010-TR-009. DoD Systems Engineering Research Center.

- Caldwell, B., & Mocko, G. M. (2008). Product Data Management in Undergraduate Education. In *Proceeding of the ASME International Conference on Computers and Information in Engineering* (pp. 433–441).
- Camba, J., & Contero, M. (2014). INTEGRATION OF MODEL-BASED DESIGN ANNOTATIONS IN PRODUCT LIFECYCLE MANAGEMENT SYSTEMS TO FACILITATE DESIGN INTENT COMMUNICATION. In *Proceedings of the 10th international symposium on tools and methods of competitive engineering*. Budapest, Hungary.
- Camba, J., Contero, M., & Johnson, M. (2014). Management of visual clutter in annotated 3D CAD Models: A Comparative Study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8518 LNCS(PART 2), 405–416. <http://doi.org/10.1007/978-3-319-07626-3-37>
- Camba, J., Contero, M., Johnson, M., & Company, P. (2014). Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability. *CAD Computer Aided Design*, 57, 61–73. <http://doi.org/10.1016/j.cad.2014.07.001>
- Chandrasegaran, S. K., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F., & Gao, W. (2013). The evolution, challenges, and future of knowledge representation in product design systems. *CAD Computer Aided Design*, 45(2), 204–228. <http://doi.org/10.1016/j.cad.2012.08.006>
- Cheney, D., & Fischer, B. (2015). Measuring the PMI Modeling Capability in CAD Systems: Report 3 - Fully-Toleranced Test Case Verification. National Institute of Standards and Technology, NIST-GCR 15-999. <http://dx.doi.org/10.6028/NIST.GCR.15-999>
- Davies, D. (2008). *Representation of multiple engineering viewpoints in Computer Aided Design through computer-interpretable descriptive markup*. University of Bath, Bath, UK.
- DoD. Department of Defense (2013). *MIL-STD 31000A: Standard Practice. Technical Data Packages*.
- Ding, L., & Liu, S. (2010). Markup in Engineering Design: A Discourse. *Future Internet*, 2(1), 74–95. <http://doi.org/10.3390/fi2010074>
- Drury, J. L., Pfaff, M. S., Klein, G. L., & Liu, Y. (2013). Decision space visualization: Lessons learned and design principles. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8007 LNCS, pp. 658–667). [http://doi.org/10.1007/978-3-642-39330-3\\_71](http://doi.org/10.1007/978-3-642-39330-3_71)
- Frechette, S. P. (2011). Model Based Enterprise for Manufacturing. In *Proceedings of the 44th CIRP International Conference on Manufacturing Systems*.
- Gunpinar, E., & Han, S. (2008). Special section: CSCWD 2006 interfacing heterogeneous PDM systems using the PLM Services. *Advanced Engineering Informatics*, 22(3), 307–316. <http://doi.org/10.1016/j.aei.2007.08.009>
- ISO. International Organization for Standardization. (2015). *ISO 16792:2015: Technical product documentation - Digital product definition data practices*.
- Kitamura, Y., Washio, N., Koji, Y., & Mizoguchi, R. (2006). Towards ontologies of functionality and semantic annotation for technical knowledge management. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 4012 LNAI, pp. 17–28). [http://doi.org/10.1007/11780496\\_3](http://doi.org/10.1007/11780496_3)
- Lubell, J., Chen, K., Horst, J., Frechette, S., & Huang, P. (2012). *Model Based Enterprise / Technical Data Package Summit Report (NIST Technical Note 1753)*. NIST Technical Note 1753.
- Mäntymäki, M., & Riemer, K. (2016). Enterprise social networking: A knowledge management perspective. *International Journal of Information Management*, 36(6), 1042-1052.
- Mun, D., & Han, S. (2005). Identification of topological entities and naming mapping for parametric CAD model exchanges. *International Journal of CAD/CAM*, 5(1), 69–82. Retrieved from <http://www.ijcc.org/ojs/index.php/ijcc/article/view/42/0>
- Park, Y., Fujimoto, T., & Hong, P. (2012). Product architecture, organizational capabilities and IT integration for competitive advantage. *International Journal of Information Management*, 32(5), 479-488.
- Pellerin, R., Quintana, V., & Rivest, L. (2013). Measuring and improving the process of engineering change orders in a

model-based definition context. *International Journal of Product Lifecycle Management*, 6(2), 138–160.  
<http://doi.org/http://dx.doi.org/10.1504/IJPLM.2012.052659>

- Quintana, V., Rivest, L., Pellerin, R., Venne, F., & Kheddouci, F. (2010). Will Model-based Definition replace engineering drawings throughout the product lifecycle? A global perspective from aerospace industry. *Computers in Industry*, 61(5), 497–508. <http://doi.org/10.1016/j.compind.2010.01.005>
- Regli, W. C., Hu, X., Atwood, M., & Sun, W. (2000). A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval. *Engineering With Computers*, 16(3-4), 209–235. <http://doi.org/10.1007/PL00013715>
- SASIG. Strategic Automotive Product Data Standards Industry Group. (2008). *JAMAEIC0422008: Strategic Automotive Product Data Standards Industry Group, 3D Annotated Model Standard*. Japan.
- Sharma, A. (2005). Collaborative product innovation: integrating elements of CPI via PLM framework. *Computer-Aided Design* 37(13):1425-1434.
- Silventoinen, A., Denger, A., Lampela, H., & Papinniemi, J. (2014). Challenges of information reuse in customer-oriented engineering networks. *International Journal of Information Management*, 34(6), 720-732.
- Soto-Acosta, P., Placer-Maruri, E., & Perez-Gonzalez, D. (2016). A case analysis of a product lifecycle information management framework for SMEs. *International Journal of Information Management*, 36(2), 240-244.
- SW. SolidWorks Connector for Aras. (2013). Retrieved April 29, 2016, from <http://www.aras.com/partners/SofTech/SolidWorks-Connector-for-Aras.pdf>
- Thouvenin, I., Guenand, A., Lenne, D., Aubry, S. (2005). Knowledge Integration in Early Design Stages for Collaboration on a Virtual Mock Up. In *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design* (Vol. 2, pp. 1141–1145). Piscataway, NJ.
- Yuan, Y.C., Fulk, J., Monge, P.R., & Contractor, N. (2010) Expertise directory development, shared task interdependence, and strength of communication network ties as multilevel predictors of expertise exchange in transactive memory work groups. *Communication Research* 37: 20-47.