ON THE APPLICATION OF MODEL-BASED DEFINITION STRATEGIES TO THE DELIVERY OF TECHNICAL TRAINING

Jorge D. Camba
Gerald D. Hines College of Architecture and Design
University of Houston
Houston, Texas, 77204
Email: jdorribo@uh.edu

Manuel Contero
I3B
Universitat Politècnica de València
Valencia, 46022, Spain
Email: mcontero@upv.es

David Pérez-López
I3B
Universitat Politècnica de València
Valencia, 46022, Spain
Email: dapelo@i3b.upv.es

Pedro Company
Institute of New Imaging Technology Universitat Jaume I
Castellón de la Plana, 12071, Spain
Email: pcompany@uji.es

ABSTRACT

   The application of computer technology to engineering and manufacturing domains has drastically transformed the way products and systems are designed and produced. However, a major drawback of CAD/CAM/CAE systems is the steep learning curve required to understand and master their extensive and increasingly complex set of functionalities. In this paper, we present a new approach to deliver CAD training materials that is inspired by Model-Based Definition (MDB) strategies, where annotated 3D models become the center of the training process. In our system, textual 3D annotations are connected to a Product Lifecycle Management (PLM) system to provide access to interactive video tutorials which are linked to specific features of a CAD model. As a proof of concerto to validate this approach, a plugin for a commercial CAD package was developed that enhances the functionality of standard 3D annotation mechanisms and enables users to interact with the technical training materials directly within the CAD interface. New data structures were implemented to support the connection and integration with PLM systems. A group of tutorials are described to illustrate the system architecture and implementation details.

Keywords: Model-based Definition (MDB), Technical Training, 3D Annotations, Product Lifecycle Management Systems (PLM).

INTRODUCTION

Modern product development largely depends on the efficient use of Computer-Aided Design, Manufacturing, and Engineering (CAD/CAM/CAE) applications. Despite the significant impact these tools have had on reducing design time and increasing the quality and competitiveness of products [1], proper training of novice users in the use of these packages still requires major investments both in terms of time and cost. Furthermore, constant software updates as well as the ongoing releases of new tools demand continuous training, even for the most experienced users [2].

Novice user training associated with Computer-Aided Design (CAD) poses unique challenges for both trainers and trainees [3]. It has been shown that traditional CAD training strategies such as lectures, workshops, and seminars, may not always be effective, particularly in large groups, due to a lack of steady feedback which can negatively affect user engagement and involvement. Collaborative approaches such as group assignments, discussions, and debates can promote creative and critical thinking, but they present similar drawbacks. Most of these problems also prevent the continuous learning of experienced users.

With ever increasing work demands placed on the engineer and less time to learn, self-directed training materials such as multimedia and video tutorials are rapidly emerging, as they provide a more active and self-paced approach to learning [4]. Research has shown that the more senses that are involved in the learning process the more effective the learning experience [5]. From a student standpoint, a wide variety of educational material is available and can be easily accessed online. For instructors, these media provide a means to distribute custom instructional materials at a low cost and with relatively no effort.

Computer-based training courses offered by CAD vendors often focus on the specific interface and workflows of their particular programs (i.e. declarative knowledge). Declarative command knowledge is particular to every CAD system, but strategic knowledge, which is associated to the election of the proper modeling strategies in a particular design situation, is much more generic and applicable across different platforms. It is also more difficult to teach and learn. For example, DS Solidworks® offers a complete curriculum in its official website, where more than a thousand pages of resources including instructor and student guides, sample projects, tutorials, and evaluation tools help users understand a variety of topics related to the application. The material is organized in four levels: middle school, high school, technical/vocational, and university/college [6]. Teaching resources include video tutorials, teacher blogs, forums, and curriculum guides. However, some authors believe the previous strategy is insufficient as many learning aspects still rely on traditional materials and lack interactivity [2].

A general roadmap of CAD training was presented by Bodein et al., [7]. In their work, the authors recommend four training steps to reach the goal of parametric design efficiency:

- Training in CAD standards and CAD rules, to learn compliance of CAD models with the PLM system and process.
- Product-oriented methodology CAD training.
- Training on templates, template experts, generic models and application knowledge-based engineering.
- Data management in PLM training.

Similarly, the standard ANSI Z490.1 [8] defines guidelines to develop training materials that are correct, consistent, effective, and well developed and delivered. The standard lists key elements that should be considered in any training program:

- Training development, including needs assessment
- Learning objectives
- Course content and format
- Resource materials
- Criteria for course completion

In this paper, we address the first and third points of the list provided by Bodein et al. [7] by developing a system to improve the delivery of video tutorials. Our system covers the creation, distribution, and use of technical materials, which can be applied to instruction once a formal training program is in place.

Our strategy consists in transforming the standard Model-Based annotation structures that are available in commercial CAD packages into mechanisms to deliver interactive training. Model-Based Enterprise (MBE) is an emerging development paradigm where the Technical Data Package of a product (3D models, PMI, performance requirements, documentation, packaging information, and other details) is used as the main source of documentation and communication throughout the product lifecycle [9, 10]. In a highly globalized world where digital information is crucial, many organizations are moving away from technical drawings to model-based approaches. Although practical challenges exist, this move represents an opportunity to work more efficiently, increase performance, and ultimately remain competitive [11-13].

Our system takes the form of a plug-in that can be integrated into an existing CAD package to connect and interface with Product Lifecycle Management (PLM) environments. Existing Product Manufacturing Information (PMI) structures used in the CAD application are synchronized with external elements stored in a PLM server. These elements may include links to external files or multimedia resources. All the information can be delivered to and accessed from the CAD application via the plug-in interface, which includes a media player, a pdf viewer, and a web browser. Support for other file formats and media can easily be included.

In the next section, we describe the architecture and implementation details of the proposed system as well as its integration with the PLM server. We elaborate on how our strategy can be used to generate, distribute, and access interactive learning materials. Finally, a hardware/software setting and workflow are proposed for an effective use of the tool.

SYSTEM ARCHITECTURE

Structure

Our system builds on the idea of enhancing existing PMI modules available in modern CAD packages by providing an additional layer of information and new functionality such as filtering, searching, and lifecycle management. The major components of the system include an annotation manager and a PLM module.

PMI annotations that are stored internally within the CAD file are automatically synchronized with enhanced external structures stored in a separate repository. The two annotation representations are linked by a unique identifier created by the PMI module. The external repository stores an xml file that contains a set of elements that describe the annotation. These elements are:

- *Feature*, refers to the specific geometric element in the annotated CAD model.
- *Text*, is the annotation content in textual form.
- *Creator*, contains the name of the creator of the annotation
- *Date*, is the date of the last modification of the annotation
- *InternalId* and *PersistentId*, represent the unique annotation identifier
- *posX, posY,* and *posZ,* represent the point in 3D space where the annotation is located
- *Website*, contains a hyperlink to a site, video, or pdf file connected to the annotation
- *Likes*, is a base64 encoded text which contains a list of users who considered the annotation useful
- *Dislikes*, same as the previous field, but representing users who considered the annotation not useful.
- *Deleted*, indicates if the annotation was removed from the model
- *AnnotationHistory*, records annotation changes. It saves the original text before modification (*text*), the date when it was modified (*modificationDate*), and the user information (*creator*)

Every CAD file is linked to an xml file in the PLM, where each annotation in the model is represented as a node, as shown in Figure 1.

```xml
<?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>0880270000000030000000002552542550000000000000000025525500100001100</persistentId>
        <posX>-0,0358368457431386</posX>
        <posY>0,0069856749169721</posY>
        <posZ>0,01417</posZ>
        <website>info.pdf</website>
        <deleted>False</deleted>
        <likes>YWRtaW4yYI0AlI2FkbWluMiNAJSNhZG1pbjIjQCUjYWRtaW4yYI0AlI2FkbWluMw==</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>
```

FIGURE 1. SAMPLE XML FILE OF ANNOTATIONS

The system ensures the integrity of the CAD model and its associated xml file in the PLM server. There is no need to manually upload or download annotations, as a background process automatically synchronizes the information. The xml

file facilitates annotation information management when working offline (without being connected to the PLM server). Changes can be updated on demand when the user finishes the modeling tasks.

The architecture of the system is based on a core module that communicates with a set of secondary modules which manage specific tasks involving the annotations stored in the CAD model and the PLM server. The functional block diagram of the proposed architecture is illustrated in Figure 2.
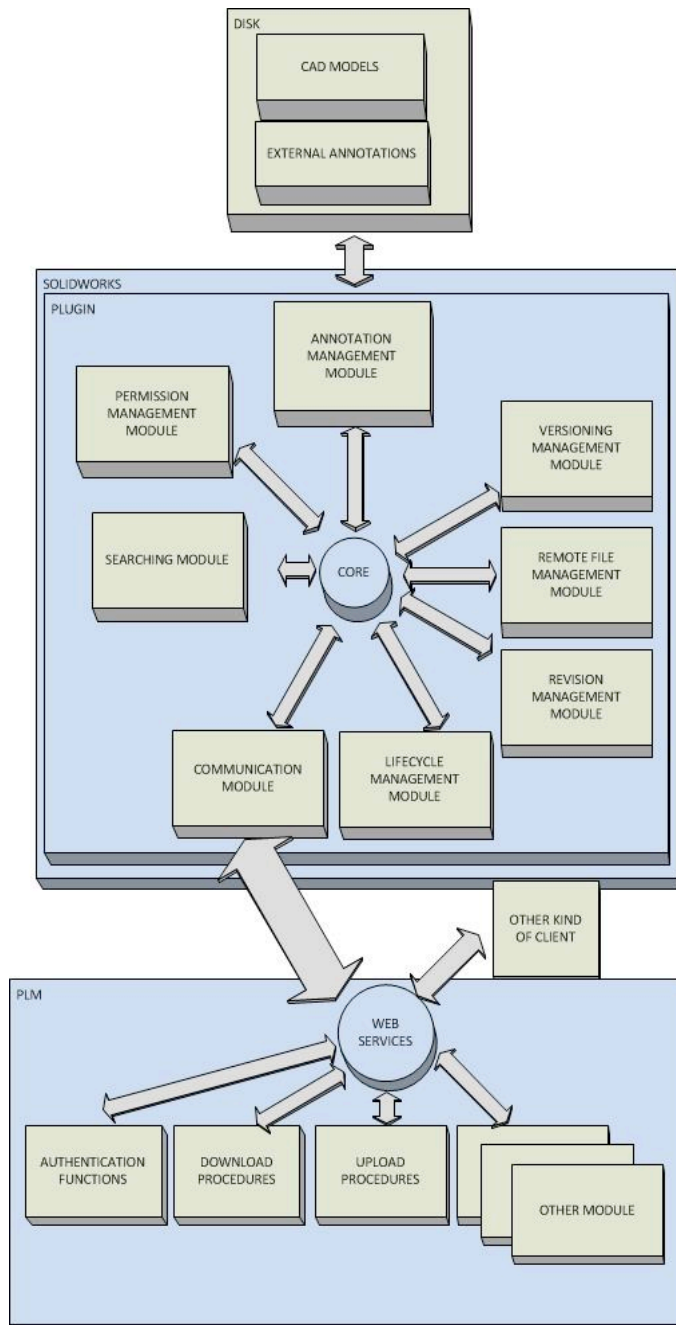

FIGURE 2. FUNCTIONAL BLOCK DIAGRAM OF PLUGIN

The core module acts as a hub for the interconnection and synchronization of the secondary modules. Each module of the system manages a specific type of data, builds queries that will be executed by the communication module, and receives PLM server responses from the core module. The core module is responsible for collecting and forwarding responses from

the communications module to the corresponding module, and vice versa. In this scenario, the PLM server is treated as a set of web services that enable clients to authenticate, "upload" (check in, set lifecycle state, set revision, lock/unlock data, etc.), and "download" (check out, get revision, get version, etc.) data.

Management of CAD models and disk operations are performed by the annotation management module whereas PLM operations are handled by the remaining modules. The permission management module collects authentication data, and establishes accessibility and permissions based on user profiles which are defined in the PLM server. The remote file management module performs *check-in* and *check-out* operations by transferring CAD models and annotations to and from the PLM server and the local machine. The versioning and revision modules control file versions and revisions respectively, allowing users to set new file versions/revisions or recover old file versions/revisions. The lifecycle management module is designed to recover and update data related to the document status. Finally, the search module facilitates user searches in the repository according to different criteria.

Implementation

The CAD package SolidWorks was selected for implementation of the prototype system. A custom plug-in was developed using Visual Basic .NET. The plug-in interface was designed as an additional tab in the SolidWorks main interface panel, as described in our previous work [14]. Model annotations are displayed in tabular form. Columns include: feature (specific part of the model where the annotation is connected to), text annotation, author, modification date (last modification date of the annotation text), link to site, video or pdf file, internal id, number of likes and number of dislikes. The table format provides an intuitive structure to browse information and apply filters. Custom filters can be defined from the plug-in interface by selecting annotation rows. Filters can be combined for more accurate results when searching large volumes of information, as shown in Figure 3.

Changes related to a particular annotation can be accessed via the History tab, which can be useful for archiving, tracking, and data mining purposes. An additional tab (Browser) implements various viewers (web browser, pdf viewer, and video player) for external media resources, so information can be viewed without leaving the CAD environment. This functionality provides the basic framework to deliver training materials using annotated models.
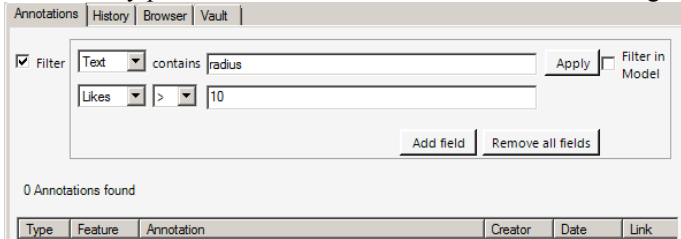

FIGURE 3. EXAMPLE OF COMBINED FILTERS

PLM integration was implemented over Aras Innovator v11, an open source PLM solution that is gaining significant attention in industry. Although relational databases were explored in our previous work [15], the proprietary nature and lack of standardization of many PLM systems called for a more flexible and open solution.

Aras Innovator uses Microsoft's Internet Information Server and an SQL database server infrastructure for storage, fast indexing, and search of data vaults. The system has three main components: a database, a web server, and a file storage system. The database server stores file information (e.g., file edit history) in the file storage system, which is implemented as a hierarchy of folders where the physical copies of the files are stored. Finally, the web server provides a set of services to manage the communication between the clients and the PLM.

Number	Name	CAD_Do...	ann_text	Parent_Directory	Feature	Creator	Date	Likes	Dislikes	cad_from [...]
										...
00DEFB...	motor.sldprt	89DB400...	video2	$/r4/	Not attach...	Dapelo	06/04/2016...	0	0	Dapelo@#$Dap...
011531...	Loreal Bottle.sldprt	4F16B61...	0-Load pics to ...	$/tutoriales/	Not attach...	Dapelo	23/09/2016...	0	0	Dapelo@#$Dap...
067258...	Squeeze Bottle.sldprt	27B998A...	4-Draw point	$/tutoriales/	Not attach...	Dapelo	20/09/2016...	0	0	Dapelo@#$Dap...
0F2E80...	Wastebasket.sldprt	16268CC...	10-Mirroring	$/tutoriales/	Mirror2	Dapelo	26/09/2016...	0	0	Dapelo@#$Dap...
133E15...	Squeeze Bottle.sldprt	27B998A...	10-Fillet	$/tutoriales/	Not attach...	Dapelo	20/09/2016...	0	0	Dapelo@#$Dap...
159CEA...	básico.SLDPRT	1312006...	4-Modify extru...	$/tutoriales/	Saliente-Ex...	Dapelo	28/07/2016...	0	0	Dapelo@#$Dap...
1A4095...	básico.SLDPRT	1312006...	3-Select the e...	$/tutoriales/	Not attach...	Dapelo	29/07/2016...	0	0	Dapelo@#$Dap...

FIGURE 4. EXTENDED ANNOTATIONS AS STORED IN ARAS INNOVATOR DATABASE

The Aras application framework uses run-time model-based technology to define and manage business objects [16]. The client side is web-based, although it provides an Application Programming Interface (API) to develop custom applications. 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 [17].

In our implementation, the CAD model, its external annotations and all associated media are uploaded to the remote repository when the CAD file is added to the PLM server. Although annotations are physically stored in a file located in the PLM server, the annotation structure is not directly accessible by the web services. To solve this problem, a new data type (*AnnotationItemType)* was created to store all the information related to the annotation as well as a reference to the corresponding CAD Document, as shown in Figure 4. By using this solution, multiple searches can be combined.

In terms of PLM functionality, a version control system keeps track of the files stored in the server as well as their lifecycle states and revisions. Before new files are uploaded to the system, information items related to the previous version are updated to maintain consistency. Older items remain in the repository and can be retrieved by selecting the corresponding version of the CAD Document.

A basic lifecycle data type was defined with states: "Work in progress," "For review," "Released," and "Obsolete." Permission to add or edit product lifecycle states is only available to PLM administrators, but state changes can be performed by any user. Revisions that mark relevant milestones or changes to a document are processed automatically by the system when the state changes to "Released" or "Obsolete."

Finally, to download a model, the user must "CheckOut" the file, which can only be completed with the appropriate permissions and when the model has not been checked out by any other user. The "CheckOut" function locks the file in the repository and downloads a copy to the local work directory. Similarly, the upload function, or "CheckIn," can only be performed if the user has the appropriate permissions and the model has been previously checked out. The operation saves all changes, adds user comments, uploads the file, the annotations, and the related multimedia and unlocks it in the server.

Interface

To simplify the implementation of the proof of concept, the interface developed in our previous work [18] was further refined in preparation for the delivery of training materials. The new interface provides combined filters, table sorting, multimedia support, and PLM connectivity.

The interface of the plug-in is divided into four tabs: annotations, history, browser, and vault. Model annotation information is displayed in tabular form in the "annotations" tab, as described earlier. When a table row is selected, the corresponding annotation is highlighted in the model. If a link to an external resource is defined, the user can view the resource directly in the "browser" tab, which implements a fully functional web browser based on Microsoft Internet Explorer, a multimedia player based on Microsoft Windows Media Player, and a pdf viewer based on Ghostscript. The tab also incorporates a simple voting system (likes/dislikes) where users can communicate whether or not they find a particular annotation useful. The "History" tab is used to display information regarding the history and changes performed to an annotation. Finally, interaction with a PLM server occurs from the "Vault" tab, where users can authenticate themselves, manage remote files, and search for annotations or files. Files are displayed in tabular form where each row represents a file version and the columns indicate version number, document revision code, document lifecycle state, author, upload date, comments, and annotation likes and dislikes.

ANNOTATED MODELS FOR TRAINING

An annotation can be defined as a comment or text block connected to an information object. In the CAD context, a model annotation refers to information, in any format (text, image, audio, and video), inserted into a CAD model and connected to a specific geometric element [15, 19]. These annotations can be used to document the modeling process in a similar way to source code comments in a computer program. The additional layer of information provided by the annotations facilitates the understanding and reusability of the model [14]. Furthermore, when properly and sufficiently annotated, comprehensive design documentation could be generated automatically from models and assemblies.

Based on the system described earlier [18], we propose the use of annotated models as facilitators of training materials. Instructors can develop educational materials by building a model step by step and adding annotations with standard PMI tools. Annotations are then linked to specific multimedia resources that can be stored in the PLM server or accessed via web. From a student standpoint, the multimedia content is delivered to the CAD environment and experienced interactively along with the annotated 3D model (see Figure 5). In this example, the video tutorial shown in the right side of the screen is explaining the modeling of the feature indicated by annotation 11 (the one selected and highlighted in the viewport).

In order to deliver a self-directed tutorial, the multimedia resources interact directly with the geometry of the model. For example, when selecting an annotation, a live 3D model can automatically revert back to the modeling step that corresponds to the part of the video the user is watching, which provides a unique approach to navigate and assess work in progress by focusing on the modeling operations that are currently active. In addition, the user can manipulate the
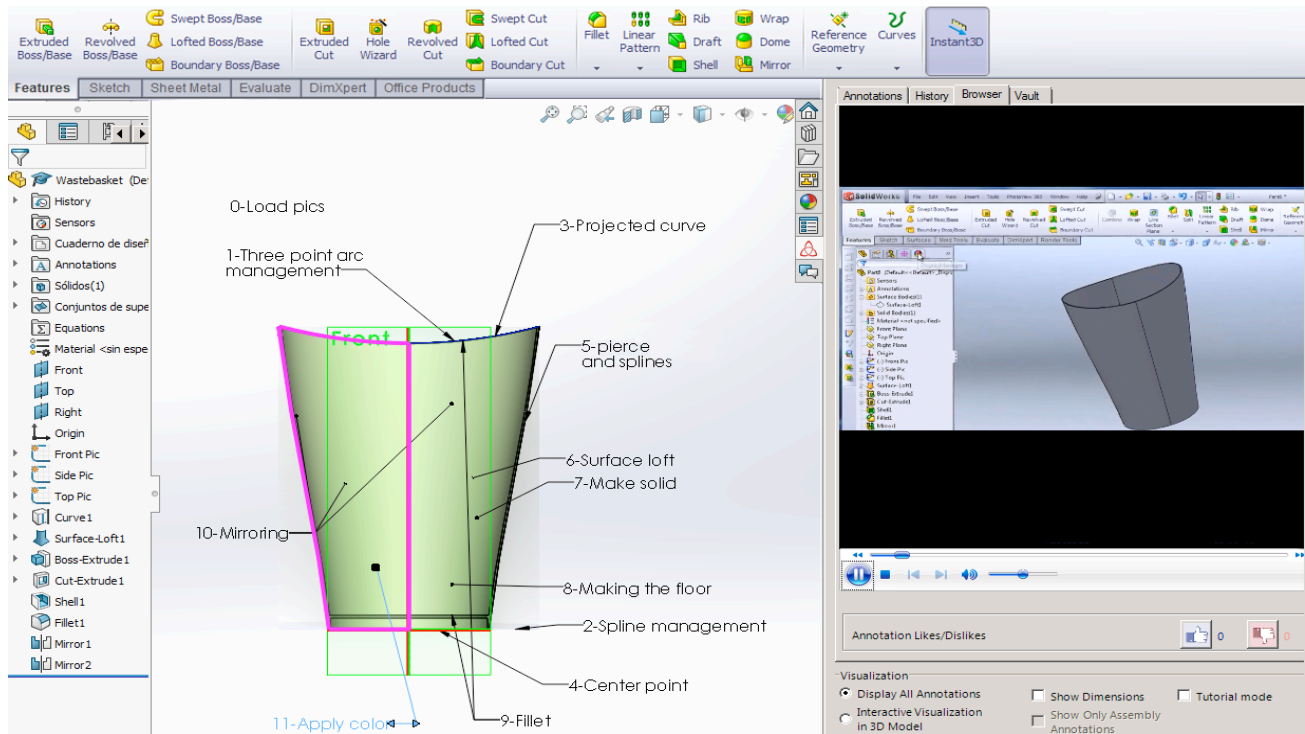
FIGURE 5. SOLIDWORKS VIEWPORT WITH ANNOTATED 3D MODEL AND VIDEO FILE LINKED TO SELECTED ANNOTATION.

annotated 3D model as it is being built, which can be beneficial in various situations, particularly when working with complex geometry that may have details that are hidden from certain angles.

The general process to create an annotation-based tutorial with our system is described. First, a video tutorial is split into multiple parts based on the annotations that will be included in the model. It is good practice to name files properly (e.g., by adding a prefix or an index to the file name) so they can be easily located in the PLM server, if necessary. Next, the instructor will create and annotate the 3D model according to her own criteria or teaching style. The multimedia resources can then be linked to the annotations at different points within the modeling process, so they can be displayed correctly.

Distribution and use of the interactive tutorials is managed by the PLM server, where the instructor can upload the resources and enable user access. Permissions can be configured by groups (i.e., only a specific group of users can access the materials) and by level (i.e., based on difficulty or user expertise). If a change or a correction needs to be made to the tutorial, the instructor can resubmit a new version to the PLM, which ensures that students will always receive the most current version of the material.

To complete a tutorial, students will launch the CAD package (e.g., SolidWorks) and use the custom plugin to login to the PLM server via the "Vault" tab. Once logged in, they will select the tutorial, which will be copied to their local machine along with all associated media. This "read-only" approach protects the original resource against accidental changes that may affect all users. The basic steps to create and distribute content are described as follows:

1. The instructorselects the video file that contains the desired video tutorial
2. The instructor edits the video file by trimming the desired learning modules into multiple parts.
3. The instructor builds and annotates the 3D model and links the annotations to the corresponding external video file.
4. The instructor checks-in the tutorial to the PLM server and defines user permissions for that particular file.
5. Students can now open their CAD system and connect to PLM server.
6. Students download the desired tutorial as read-only files.
7. Students complete the tutorial by directly interacting with the annotated model and the video file.

The search functionalities implemented for model annotations in the original system can also be used with tutorials. For example, a student may be interested in finding tutorials related to a specific CAD file and created by a particular instructor (see Figure 6). Similarly, the voting mechanism used to rank annotations (likes/dislikes) can also be used to rank tutorials, which adds a social dimension to the learning experience. Since all tutorials all stored in the PLM server, a control

system was implemented to ensure users only vote once per tutorial. When the "like" or "dislike" buttons are pressed and the vote is submitted, the buttons will be automatically disabled in the plug-in interface for the active tutorial. The number of likes/dislikes can also be used as search criteria to identify the best tutorials or those preferred by the students. This functionality encourages student interaction and participation. It can also be used by instructors to identify tutorials that need to be reworked or clarified.

Finally, a recommended setup and workflow for a training session that uses the proposed system is described. Since users will likely be creating their own models while watching the tutorials and manipulating the corresponding annotated models from the PLM server, a widescreen monitor is suggested, where both models can be displayed simultaneously at reasonable sizes. A dual monitor workstation is preferred. Such wide workspace is required as three elements will need to be displayed simultaneously at all times: the video player window and the two 3D models.

For practical reasons, the two document windows of the 3D models in the CAD package (SolidWorks) interface need to be arranged side by side. The plug-in panel with the video player tab can be pinned to the interface so it will be always visible. In this workspace setup, the user can move freely from one element to the next without having to switch applications.

FIGURE 6. SEARCHING TUTORIALS USING COMBINED FILTERS

CONCLUSIONS AND FUTURE WORK

Modern approaches to product development rely on digital CAD models to drive nearly all aspects of the product lifecycle. Despite the widespread use of CAD in engineering design fields and the increasing popularity of the Model-Based Enterprise (MBE) paradigm, mastery of these programs is an intellectually demanding and time-consuming task.

In this paper, we presented a practical approach for delivering strategic knowledge which is triggered by queries or incorrect actions of the users. Training materials and resources are connected to specific aspects of the geometry of a 3D model via annotations. The system relies on annotated CAD models as carriers of training materials. It is designed as a student-centered learning platform where instructors can create and distribute custom educational content, and students can access and complete it at their own pace in an interactive manner directly from the design environment.

The Model-Based approach facilitates the connection between 3D model and instructional materials. 3D models are transformed into interactive assets that trainees can explore and inspect in parallel while watching the corresponding video tutorials. Furthermore, work in progress can be examined and potentially compared to the student's work at various modeling stages for continuous assessment.

A working prototype of our system was implemented as a plug-in for a specific CAD package to initiate formal validation and a recommended hardware-software setup and workflow were described for conducting effective training sessions. As future work, we plan to evaluate both the usability of the tool and its effectiveness in an educational setting. We are particularly interested in comparing our strategy with standard video tutorials and other traditional CAD training methods. In addition, we are also interested in studying whether the proposed approach could provide benefit to industry in terms of streamlining the way the workforce is trained in using CAD systems. Further studies will also include potential applications beyond CAD training, such as the development and delivery of interactive manufacturing/quality work instructions and interactive technical publications. Training for non-technical applications such as visualization and digital media will also be explored.

ACKNOWLEDGEMENTS

REFERENCES

[1] Leu, M.C., ElMaraghy, H.A., Nee, A.Y., Ong, S.K., Lanzetta, M., Putz, M., Zhu, W. and Bernard, A., 2013. "CAD Model Based Virtual Assembly Simulation, Planning and Training," CIRP Annals-Manufacturing Technology, **62**(2), pp.799-822.

[2] Field, D.A., 2004, "Education and Training for CAD in the Auto Industry," Computer Aided Design, **36**(14), pp 1431-1437.

[3] Hamade, R.F., Artail, H.A. and Jaber, M.Y., 2005, "Learning theory as applied to mechanical CAD training of novices." International Journal of Human-Computer Interaction, **19**(3), pp.305-322.

[4] James-Gordon, Y. and Bal, J., 2003, "The emerging self-directed learning methods for design engineers." The learning organization, **10**(1), pp.63-69.

[5] Barraclough, A., and Guymer, I., 1998, "Virtual Reality - A Role in Environmental Engineering Education?" Water Science and Technology, **38**(11), pp. 303-310.

[6] Solidworks Curriculum and Lessons, 2016, http://www.solidworks.com/sw/education/educational-curriculumresources-3d-cad.htm

[7] Bodein, Y., Rose, B. and Caillaud, E., 2013, "A Roadmap for Parametric CAD Efficiency in the Automotive Industry," Computer-Aided Design, **45**(10), pp.1198-1214.

[8] Schroll, R. C., 2004, "ANSI Z490. 1-2001: Criteria for Accepted Practices in Safety, Health, and Environmental Training". In ASSE Professional Development Conference and Exposition. American Society of Safety Engineers.

[9] Department of Defense, 2013. "MIL-STD 31000A: Standard Practice. Technical Data Packages."

[10] 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.

[11] Frechette, S. P., 2011. "Model Based Enterprise for Manufacturing." In Proceedings of the 44th CIRP International Conference on Manufacturing Systems.

[12] 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.

[13] 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.

[14] Camba, J., Contero, M., and Johnson, M., 2014, "Management of visual clutter in annotated 3D CAD Models: A Comparative Study". Lecture Notes in Computer Science **8518**(2), pp. 405-416.

[15] Camba, J., and 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.

[16] Aras Innovator, http://www.aras.com

[17] SolidWorks Connector for Aras, 2013, http://www.aras.com/partners/SofTech/SolidWorks-Connector-for-Aras.pdf

[18] Camba, J., Contero, M., Johnson, M., and Company, P., 2014, "Extended 3D Annotations as a New Mechanism to Explicitly Communicate Geometric Design Intent and Increase CAD Model Reusability," Computer Aided Design, **57**, 61-7.

[19] Ding, L., and Liu S., 2010, "Markup in Engineering Design: A Discourse." Future Internet, **2**, pp. 74–95