Secure CAD Model Retrieval and Data Consistency: Issues in Role-based Visualization

Zhiming Qiu[1], Jerry Y. H. Fuh[2] and Yoke San Wong[3]

[1]National University of Singapore, mpeqzm@nus.edu.sg
[2]National University of Singapore, mpefuhyh@nus.edu.sg
[3]National University of Singapore, mpewys@nus.edu.sg

ABSTRACT

Application and development of collaborative design solutions is hindered by the lack of secure 3D data sharing among a wide range of participants in the product development chain. Role-based visualization facilitates presentation of CAD models at different visualization levels to intra- or inter-enterprise collaborators and thus protects confidential enterprise design data. By extending the traditional three-level (i.e. write, read, and deny) information security mechanism and defining the relationships between various roles, 3D models and visualization levels, information sharing is achieved with enhanced information security. The main purpose of the `role-based visualization' concept is to support collaboration on Product Lifecycle Management (PLM) platforms. As a continuation of previous work, this paper highlights two key issues for role-based visualization, i.e. (1) secure model retrieval and (2) data consistency. Implementation details are presented and discussions on further work are also given.

Keywords: Role-based visualization, CAD, information security, information consistency, collaborative design.

1. INTRODUCTION

Collaboration plays an increasingly significant role for manufacturers to gain a leading edge in present competitive business environment. Manufacturers are interested in effective information exchange among key participants in product development cycle, especially the feedbacks from the downstream phase to the early design phase. Since 3D CAD applications are widely adopted in product design stage, sharing 3D CAD models to downstream collaborators, such as manufacturing engineers, component suppliers and customers, can effectively enhance early detection of defects and optimize the entire product development cycle. However, it is imperative to manage confidential and proprietary design knowledge when sharing models to intra- or inter- enterprise users.

The following example illustrates a typical scenario. Manufacturer A develops product X which is represented in a CAD application as an assembly model S consisting of three components P, Q, and R. Manufacturer A out-sources components P and Q to companies B and C, respectively. To facilitate effective model sharing among collaborators while protecting key information from unauthorized access, manufacturer A establishes the following policies: (1) designers in A have full access to S, i.e., they can view and modify S as well as its linked components P, Q, and R; (2) designers in B have full access (view and modify) to P, and partial access (view) to Q and R; and (3) designers in C have full access to Q, and partial access to P and R.

Current Product Data Management (PDM) systems can meet the above requirements by enforcing a three-layer (read, write, deny) security mechanism on the 3D models, which is similar to the security mechanism adopted by UNIX operating systems. These PDM systems support either full access (view and modify) or partial access (view). If a user has partial access to a CAD model, he can retrieve the model from a PDM database to his local workspace (a local storage space or assigned space on a server) and make revisions on it (if he has the editing software); however, he cannot replace the original model with the revised one in the database.

It is obvious that a high risk is incurred when granting external users with partial access (view) since the full details of the design part can be disclosed. Based on the above consideration, many manufacturers are reluctant to adopt inter-

enterprise collaborative solutions for 3D model sharing-out; instead, they choose to limit the scope of collaboration inside their enterprises and hence cannot optimize product development in a global scale.

Therefore, it is desirable to develop technologies to deliver both a powerful and secure 3D model sharing. For example, external collaborators can access confidential design models at a relatively low level, e.g., with reduced feature sets removed and simplified tessellations. With the enhanced flexibility in model sharing, manufacturers gain more confidence to adopt collaborative solutions with their partners. Obviously, existing commercial collaborative solutions do not deliver such functionality and thus more research needs to be done to address the gap.

Role-based collaborative 3D model protection is a recent research topic. One pilot research was done by Cera et al. [1]. They proposed a co-modeling mechanism for concurrent designers. Except its authorized part, each participant can only view the remaining parts via viewing envelopes. Geometric simplification, with genus removal, if applicable, was applied to generate simplified viewing meshes. Focusing on synchronous co-modeling, however, their work is difficult to be extended in current PLM environments which feature asynchronous collaborations. To support collaborative design, Shyamsundar et al. [3] reported a geometric representation of assembly models, AREP, to enhance visualization and edition of parts in distributed environments. However, their work does not address role-based collaboration mechanism. To realize role-based visualization in current PLM systems, inspired by [1], Qiu et al. [2] proposed a similar approach and implemented it in SolidWorks/SmarTeam environment. However, their approach [2] does not address a security flaw in assembly check-out and the data inconsistency problem will result from data revision.

This paper aims to improve the approach in [2] by addressing the above two problems. The rest of the paper is organized as below: Section 2 illustrates the basic concept of role-based visualization; Section 3 and 4 address the two key issues, i.e., secure data retrieval and data consistency, respectively. In Section 5, details on implementation in a commercial collaborative environment are presented. Finally, conclusions and discussions are given in Section 6.

2. ROLE-BASED VISUALIZATION

In collaborative environments, *role* is widely used to represent a group of users with similar job responsibilities and access privileges. Since users under different roles might represent different organizations, information security policies should be carefully specified to prevent disclosure of enterprise proprietary information to external entities, even to collaborators [3][5][6][8]. Actually, the lack of flexible 3D information management prevents industries from adopting more 3D data sharing between organizations.

Role-based visualization aims to achieve reliable and flexible 3D data access control on collaborators so as to enhance the current `all-or-nothing' approach. Its main concept is to divide a whole 3D model, e.g., an assembly model, into several parts and specify an access level on each part for a role. As discussed in [1], for each model, a matrix is defined to specify the relationship between roles and parts. Each element in the matrix determines the access level for a role-part pair.

Although there are various CAD applications and modeling mechanisms, it is reasonable to define a common set of access levels: original model, original tessellation, simplified tessellation, and bounding volume. For an end user, a part represented as original model is editable while original visualizations provide non-editable models for accurate viewing. The simplified visualization and bounding volume refer to higher degree of information hiding. A simplified tessellation may be the simplified meshes or hierarchical bounding volumes.

In the previous work [2], there are four access levels - original model, tessellations with controlled quality, feature bounding box, and bounding box. Fig. 1 illustrates the four access levels. At the first level, a user can access to the original model, modify it, and save it back to the PDM database. At the second level, the user can view the model with predefined visual fidelity; however, he cannot modify it or save it back to the PDM database. Geometrical simplification [7] might be applied to generate meshes if there is no way to control the quality of meshes inside CAD applications. At the third level, the model is represented as a set of axis-aligned bounding boxes which can be nested to reflect the hierarchical dependency between features. Each bounding box represents a design feature in the original model. At the fourth level, the entire part is represented as a bounding box for maximized reduction of geometries. After being saved in a PDM database, each part is compiled into four configurations for each visualization levels. These four configurations are saved in a PDM database with the original design model. It should be noted that a

configuration is not an independent entity: it just serves as an additional visualization for the underlying part and specifies how the part is visualized. The relationship between roles and access levels is specified and saved into the corresponding assembly model. When a user requests the assembly model, the access

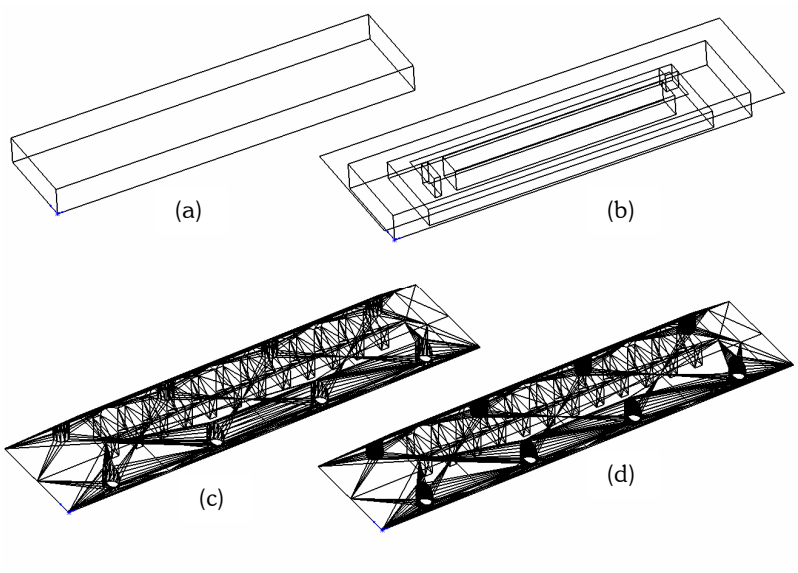

Fig. 1. Different access levels – (a) bounding box, (b) feature bounding box, (c) tessellations, and (d) original model.

levels for all parts in the assembly models are retrieved from the role-access-level relationship and parts are visualized accordingly in the entire assembly model. However these parts are visualized at their individual access levels in the assembly model; they are actually downloaded to the local workspace of a user at the full access level since configurations cannot be viewed alone without the presence of the underlying part. Once the user has the original part in his local workspace, the information protection is broken.

When a user modifies a part and save it back to a PDM database, it is his responsibility to notify the administrator to re-generate multiple-level visualizations for the part and update the role-access matrix in the database. When design changes are frequent, this manual approach is time-consuming and a more effective mechanism is needed to update visualizations and role-access matrices upon arrivals of part changes.

3. SECURE DATA RETRIEVAL

As a common policy implemented in most PDM systems, if a user can access an assembly model, he can assess all its components as well. Hence, the user can check out all parts when he checks out the assembly and thus introduces a potential security problem. To solve the problem, the parts in the assembly should be modified to an appropriate access level when the user does not have the full access to them. The following steps are needed for the above goal:

- Retrieve the visualization of a part for the user. If the user has the full access, the part is downloaded into the user's workspace as normal; else go to next step.
- A proxy of the part is initialized and all features in the part are generated in accordance with the visualization level of the part for the user.
- The entire assembly model is generated by aggregating original or proxy parts together in the user's local workspace.
- The user can modify original parts to which he has the full access
- Modified original parts are saved back to the database. All proxy parts are left alone in the user's local workspace.

There are two ways to generate proxy parts: (1) generating at the retrieving time and saving them into users' local workspace, and (2) generating in advance and caching them in PDM databases for future retrieval. The first method can save space at server side, but leads to performance overhead at both server and client sites at data retrieval times; meanwhile, the second method makes data retrieval faster but consuming more space at the server end.

As a shortcoming, the second approach does not support 'one-user-multiple-role' scenario where a user belongs to multiple roles and the visualizations for these different roles should be combined for final presentation. For example, user A is attached to two roles R1 and R2. A part P can be presented to R1 at original NURBS level, and to R2 at simplified mesh level. Here the ambiguity arises as it cannot be determined whether original NURBS or simplified mesh should be presented to user A. Hence, a policy should be set up to solve the conflict. A conservative rationale is always to present the lower visualization to users. In the above example, the mesh of the part is presented to user A.

4. DATA CONSISTENCY

It is necessary to notify the administrator of the PDM system to update access level information and access-role matrices when some parts are modified. An effective way to achieve this is to adopt a workflow since many PDM systems have built-in workflow engines. Since change involves both the user side and the administrator side, the workflow capability needs to cover both sides, too. The entire procedure takes the following steps:

- Models are retrieved out from the PDM database to the user side;
- A CAD application is launched at the user side to perform the change on the models;
- The user saves back the models to the PDM database; and
- A workflow at the server side is triggered to check the integrity and notify the administrator to manage any data inconsistency.

In the 4-step approach, the last step is the key one since the change of models on user side may introduce data inconsistency in the following ways:

- A part is added into the assembly model. Therefore, all access levels of visualization for the part should be generated, and more entries in the access-role matrix should be created to link the access levels of the part to the existing roles.
- A part is removed from the assembly model. In this case, all entries related to the part in the access-role matrix should be removed.
- A part is modified, e.g., removal, addition or modification of features. Hence, all access levels for the part should be re-generated, and the access-role matrix might be updated.

It is safe and straightforward to call the administrator via workflows or other notification mechanisms to perform the update manually. However, the above three cases should be handled separately instead of re-generating the access level and role-access matrices entirely, i.e.

- Part addition. The administrator is notified to manually suppress features, generate all visualizations, and add entries in the access-role matrix to define access levels for roles.
- Part removal. The administrator is notified to remove in the access-role matrix all entries related to the removed part.
- Feature addition or modification. The administrator re-generates all visualizations for the related part, evaluates the confidential level of the features to decide if the feature should be suppressed, and modifies in the access-role matrix the entries related to the part accordingly.
- Feature removal. The administrator re-generates all visualizations for the related part.

5. IMPLEMENTATION

A basic role-based visualization foundation is under implementation in SolidWorks/SmarTeam PDM environment as illustrated in Fig. 2. All adds-on are to be realized as macros (for SolidWorks) or scripts (for SmarTeam) in VB programming language with the support of SolidWorks and SmarTeam APIs. These adds-on facilitate functionality such as visualization (configuration) generation, role-access matrix definition, secure model retrieval, and data consistency validation. The SolidWorks macros accesses data management and workflow functions via the SmarTeam

APIs. As illustrated in Fig. 1, an assembly model includes component parts and a role-access matrix, and each part has several configurations for different visualizations.

The entire system works as below:

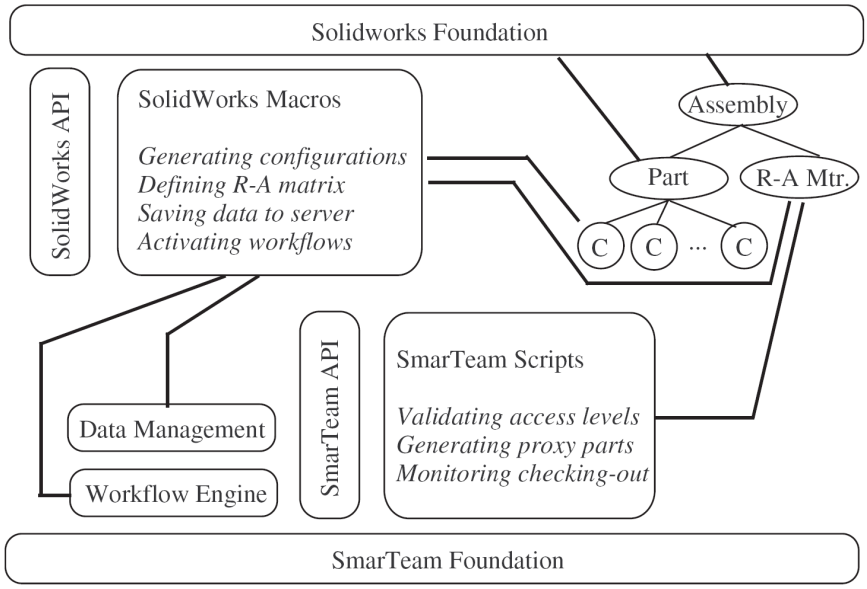

Fig. 2. Prototype system architecture.

- Designers generate parts and assembly models with SolidWorks applications and save them into SmarTeam database via SolidWorks macros.
- Administrators launch SolidWorks macros to generate all levels of visualizations for parts and link them to roles.
- When a user requests an assembly model, the script at the SmarTeam side checks the user's role against the access-role matrix for the assembly model, generates proxy parts based on the privileges of the user, and transmits them to the user's local workspace
- The user can modify parts represented in the full visualization level, save them back to the SmarTeam database, and notify the script at the SmarTeam side to update the access-role matrix and visualizations for the assembly model based on the default updating rules.
- Administrators are notified with the change and can supervise the entire updating process.

Fig. 3 illustrates visualizations of an engine assembly for different collaborators. Four roles are involved in the visualization: project manager, engine block/top front designer, engine header designer, valve cover designer, and radiator/battery designer.

6. CONCLUDING REMARKS

This paper presents approaches to facilitate secure data retrieval and consistency for role-based visualization and their implementation in commercial CAD and PDM systems. The proposed approaches are generic and not limited to any specific CAD/PDM systems if CAD and PDM systems can be integrated at the API level. At present, the implementation is based on CAD/PDM APIs to extend the functionality of the CAD and PDM applications due to the lack of a standard CAD format to manage multiple levels of visualizations. However, there are ongoing efforts addressing this need, such as the universal 3D initiative driven by Intel. With the progress on these formats, role-based visualization can be an inherent feature in major PLM environments to support 3D data sharing soon. In addition, the role-based visualization method can be extended to feature levels to offer more flexibility to the current `suppress-or-present' scenario.

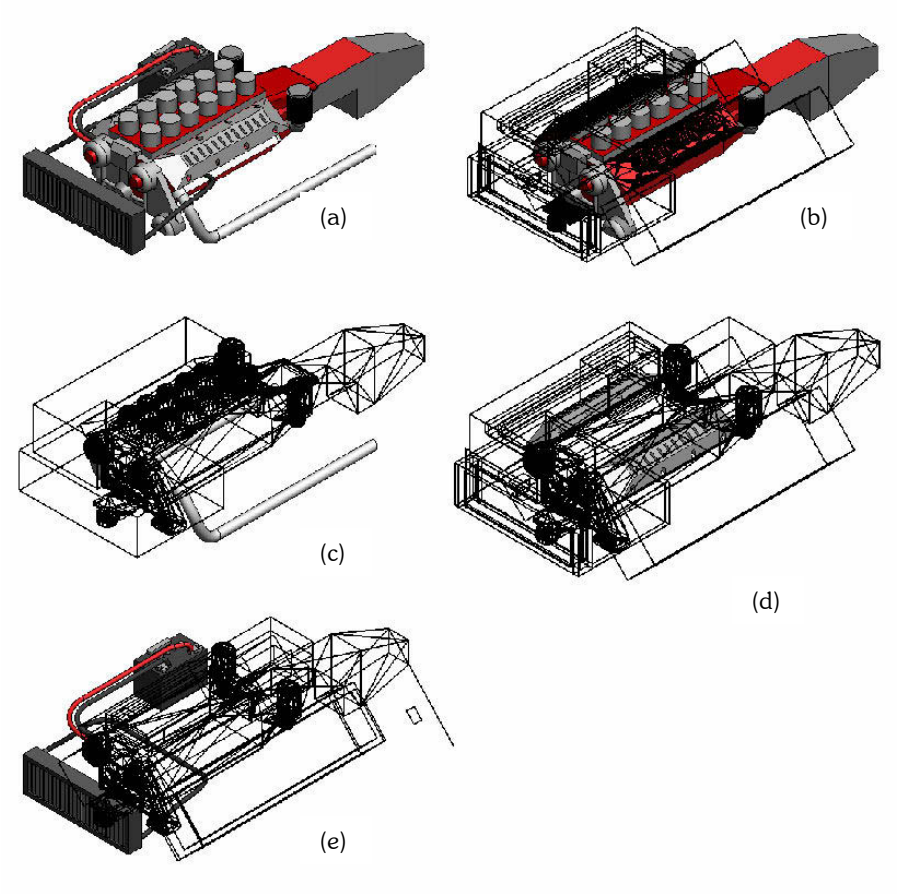

Fig. 3. Visualizations of an engine for different roles: (a) project manager (full NURBS), (b) engine block/front/top designer, (c) engine header designer, (d) valve cover designer, and (e) radiator/battery designer.

7. REFERENCES

[1] Cera, D. D., Kim, T, Han, J and Regli, W. C., Role-based viewing envelopes for information protection in collaborative modeling, *Computer-Aided Design*, Vol. 36, No. 1, 2004, pp 873-886.

[2] Qiu, Z. M., Kok, K. F., Wong, Y. S. and Fuh, Jerry Y. H., Role-based visualization in asynchronous collaboration, submitted to *Computer in Industry*.

[3] Sandhu, R. S., Coyne, E. J., Feinstein, H. L. and Youman, C. E., Role-based access control models, *IEEE Computing*, Vol. 29, No. 2, 1996, pp 38-47.

[4] Shyamsundar, N. and Gadh, R., Internet-based collaborative product design with assembly features and virtual design spaces, *Computer-Aided Design*, Vol. 33, No. 9, 2001, pp 637-651.

[5] Van der Hoeven, A. J., ten Bosch O, van Leuken R, van der Wolf P, A flexible access control mechanism for CAD frameworks, *Proceedings of the Conference on European Design Automation Conference*, IEEE Computer Society Press, 1994, pp 188-193.

[6] Stevens G. and Wulf, V., A new dimension in access control: studying maintenance engineering across organizational boundaries, *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, New York, ACM Press, 2002, pp 196-205.

[7] Luebke, D. P., A Developer's Survey of Polygonal Simplification Algorithms, *IEEE Computer Graphics and Applications*, 2001, pp 24-35.

[8] Ferraiolo, D. E., Role-based access control (RBAC): features and motivations, *Proceedings of the 11th Annual Computer Security Applications Conference*, 1995, pp 241-248.