

**Christopher D. Cera**  
**Ilya Braude**

Geometric and Intelligent Computing Laboratory,  
Department of Computer Science,  
Drexel University,  
Philadelphia, PA 19104

**Taeseong Kim**  
Mobile Multimedia Laboratory,  
LG Electronics Institute of Technology,  
Seoul, 137-724, Korea

**JungHyun Han**  
Department of Computer Science and  
Engineering,  
Korea University,  
Seoul, 136-701, Korea

**William C. Regli**  
Geometric and Intelligent Computing Laboratory,  
Department of Computer Science,  
Drexel University,  
Philadelphia, PA 19104

# Hierarchical Role-Based Viewing for Multilevel Information Security in Collaborative CAD

*Information security and assurance are new frontiers for collaborative design. In this context, information assurance (IA) refers to methodologies to protect engineering information by ensuring its availability, confidentiality, integrity, nonrepudiation, authentication, access control, etc. In collaborative design, IA techniques are needed to protect intellectual property, establish security privileges and create “need to know” protections on critical features. This paper provides a framework for information assurance within collaborative design based on a technique we call Role-Based Viewing. We extend upon prior work to present Hierarchical Role-Based Viewing as a more flexible and practical approach since role hierarchies naturally reflect an organization’s lines of authority and responsibility. We establish a direct correspondence between multilevel security and multiresolution surfaces where a hierarchy is represented as a weighted directed acyclic graph. The permission discovery process is formalized as a graph reachability problem and the path-cost can be used as input to a multiresolution function. By incorporating security with collaborative design, the costs and risks incurred by multiorganizational collaboration can be reduced. The authors believe that this work is the first of its kind to unite multilevel security and information clouded with geometric data, including multiresolution surfaces, in the fields of computer-aided design and collaborative engineering. [DOI: 10.1115/1.2161226]*

## 1 Introduction

Information assurance (IA) refers to methodologies to protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation. In collaborative design, IA is mission-critical. Suppose a team of designers is working collaboratively on a 3D assembly model. Each designer has a different set of security privileges and no one in the team has the “need to know” the details of the entire design. In collaboration, designers must interface with others’ components/assemblies, but do so in a way that provides each designer with only the level of information he or she is permitted to have about each of the components. For example, one may need to know the exact shape of some portion of the part (including mating features) being created by another designer, but not the specifics of any other aspects of the part. Such a need can also be found when manufacturers outsource designing a subsystem: manufacturers may want to hide some critical information of the entire system from suppliers.

The authors believe that a geometric approach to IA represents a new problem that needs to be addressed in the development of collaborative CAD systems. The approach we develop has many uses visible across several significant scenarios we envision for applying this work:

**Protection of sensitive design information:** As noted above, designers may have “need to know” rights based on legal, intellectual property, or national security requirements.

**Collaborative supply chains:** Engineering enterprises outsource a considerable amount of design and manufacturing activity. In many situations, the organization needs to provide vital

design data to one partner while protecting the intellectual property of another partner.

**Multidisciplinary design:** For designers of different disciplines working on common design models, designers suffer from cognitive distraction when they must interact with unnecessary design details that they do not understand and cannot change. For example, an aircraft wheel well [1] is a complex and confusing place in which electronics, mechanical, and hydraulics engineers all must interact in close quarters with vast amounts of detailed design data. These interactions could be made more efficient if the design space could be simplified to show each engineer only the details they need to see.

This paper develops a new technique for Role-Based Viewing [2] in a collaborative 3D assembly design environment, where multiple users work simultaneously over the network, and presents a combination of *multiresolution geometry* and *multilevel information security models*. Among various issues in IA, *access-control* is critical for the purpose. We demonstrate the specification of access privileges to geometric partitions in 3D assembly models defined based on the Bell-La Padula model. Aside from digital 3D watermarking, research on how to provide IA to distributed engineering teams, working in collaborative graphical environments, remains a novel and relatively unexplored area. We achieve these functional capabilities within a system designed for secure, real-time collaborative viewing of 3D models by multiple users working synchronously over the internet on standard graphics workstations. The contributions of this work, developed in [2], include:

Providing a geometric approach to Information Assurance: Our work augments currently practiced access-control techniques in collaborative CAD and PDM systems. Although most of these systems offer access-control facilities, they are often limited to prohibiting access to models and documents and not partitions of geometry.

Developing alternatives to the problem of “all-or-nothing” per-

Contributed by the Engineering Simulation and Visualization of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received January 26, 2004; final manuscript received March 24, 2005. Assoc. Editor: N. Patrikalakis.

missions: The standard method for handling a lack of appropriate permissions is suppression of the sensitive features. This work attempts to highlight some alternatives other than the traditional solution.

In our revised method, we show how geometric partitions are used to create multiple level-of-detail (LOD) meshes across parts and subassemblies to provide a *Role-based View* suitable for a user with a given level of security clearance. The specific contributions of this paper include:

**Introducing role hierarchies:** We revisit the problem of role-based viewing in an updated context using role hierarchies. A hierarchy is represented as a weighted directed acyclic graph (DAG), and the permission discovery process is formalized as a graph reachability problem.

**Outlining the relation between multilevel security (MLS) hierarchies and multiresolution surfaces:** In [2] we introduced a *multiresolution envelope* to degrade surface features for a sensitive part or subassembly where details need to be withheld. We incorporate this approach into role-hierarchies where the path-cost is used as input to a multiresolution function. The authors believe that this work represents a unique application of multiresolution surfaces to multilevel information security in computer-aided design and collaborative engineering.

This paper is organized as follows: Sec. 2 describes related work from information assurance, collaborative design, and computer graphics communities. Section 3 first reviews the specification of security features in the fields of solid modeling and engineering as outlined in Ref. [2], then presents *Hierarchical Role-Based Viewing*. Section 4 explains the details of our multiresolution security model and outlines its relation to the *Role Hierarchy*. Section 5 describes the implementation of our prototype system, and demonstrates a sample scenario using our approach. Lastly, Sec. 7 summarizes our results, presents our conclusions, and outlines goals for future research.

## 2 Related Work

The contributions presented in this paper are related to information assurance, collaborative design, and multiresolution surface generation.

**2.1 Information Assurance and Security.** Current research on information assurance incorporates a broad range of areas focused on protecting information and information systems by ensuring their availability, integrity, confidentiality, nonrepudiation, authentication, and controlling modes of access. Information assurance research, in the context of the CAD domain, has been partially addressed by the computer graphics community through the development of 3D digital watermarking [3]. Digital Watermarking is used to ensure that the integrity of a model has been maintained, as well as provide a foundation for proof of copyright infringement. Other areas of research have been in authentication and access-control. We will introduce past and present research on access control methodologies and outline the differences between the varying policies.

There is a clear distinction between authentication and access control services. Authentication services are used to correctly determine the identity of a user. Access control is the process of limiting access to resources of a system only to authorized users, programs, processes, or other systems. Authentication is closely coupled with access control, where access control assumes that users of an information system have properly been identified by the system. If the authentication mechanism of a system has been compromised, then the access control mechanism that follows will certainly be compromised. The primary focus of our work is to articulate an access control policy, specifically for the geometry of a solid model, assuming a robust authentication mechanism has already been established. Access-control literature describes high-level policies on how accesses are controlled, as well as low-level mechanisms that implement those policies.

The common access control policies found in literature are Discretionary, Lattice-Based, and Mandatory Access Control (DAC, LBAC, and MAC, respectively). DAC was formally introduced by Lampson [4], where essentially the owner of an object has discretion over what users were authorized to access that object. Access broadly refers to a particular mode of operation such as read or write. The owner is typically designated as the creator of an object, hence it is an actual user of the system. This is different from LBAC and MAC, which we will refer to collectively as MAC [5], where individual users have no discretion over object access. MAC [6] is primarily concerned with the flow of information, thereby enforcing restrictions on the direction of communication channels. For further discussion on access control policies, we refer interested readers to a survey by Sandhu [7].

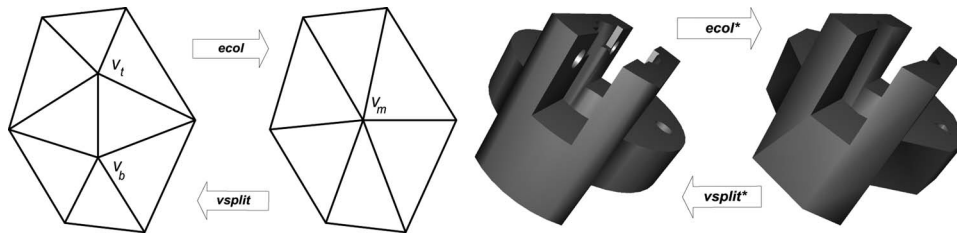
Role-Based Access Control (Group policies found in current operating systems could be viewed as an instance of hierarchical RBAC). RBAC is an emerging area of study, and is actively pursued as an augmentation of traditional DAC and MAC. RBAC is a type of Multi-Level Security (MLS) framework, which is an actively pursued area in the database community [8,9]. In RBAC, individual users are assigned to roles, and the access permissions of an object are also assigned to roles. Therefore the permissions assigned to a role are acquired by the members associated with it. This additional layer reduces the management of permissions and supports the concepts of least privilege, separation of duties, and data abstraction. RBAC, and its associated components, are an instrument for expressing a policy, and not a policy by itself. For *role-based viewing*, we use a MAC policy embodied within an RBAC framework.

**2.2 Collaborative Design.** There is a vast body of work on concurrent engineering and distributed collaborative design. Visualization and multiuser protocols were the primary focus of many early systems, and generally fell under the terms Distributed and/or Collaborative Virtual Environments [10–17]. Thereafter, lightweight design [18–21] and assembly modeling systems [22] for distributed CAD began to emerge.

With the demand for distributed CAD and more edit-driven modeling mechanisms, several research systems and commercial products have reached fruition. Current work in distributed collaborative design, with respect to geometry, can be “loosely” grouped into two categories: visualization and annotation of CAD models; co-design and manipulation of CAD geometry. The current demonstration of our work is primarily targeted at the former category. This is complementary to facet-based editing techniques [23], however editing multiresolution geometry hierarchies can be extended to feature, assembly, and native-geometry editing facilities. This assumes a system with policies that permit editing on a low resolution model, where the user in question does not have permissions to view the full resolution model. Polyinsantiation issues are beyond the scope of this paper, although most theoretical issues have been developed with regard to databases [8,9].

The generation of facets is inevitable in the visualization process, and most distributed CAD systems deal with this problem in very diverse ways. System designers must resolve varying trade-offs in the areas of visualization, editing mechanisms, bandwidth utilization, and accuracy. Some systems have targeted view-dependent techniques as an attempt to reduce aggregate bandwidth and computation [24–27]. Current literature now provides numerous mechanisms and interfaces from which to edit CAD models: features [28,29]; assemblies [30,31]; native surfaces (e.g., b-rep) [32,33]; and facet-based techniques [23]. The heterogeneity of existing systems and interchange formats further complicates the domain. Reference [28] contains an extensive overview of current distributed CAD systems from the perspective of architecture and data exchange.

**2.3 Multiresolution Techniques.** Polygon meshes lend themselves to fast rendering algorithms, which are hardware-accelerated in most platforms. Most CAD models are represented



**Fig. 1** Illustration of multiresolution techniques. (a) Edge collapse (*ecol*) and vertex split (*vsplit*) operations.  $v_t$  (top) and  $v_b$  (bottom) collapse into  $v_m$  (middle). The inverse operation involves splitting  $v_m$  back into  $v_t$  and  $v_b$ . (b) Sequence of operations on the “socket” model [37].

as a set of trimmed parametric surfaces, so tessellation to a desired resolution is performed [34,35]. Many applications, including CAD, require highly detailed models to maintain a convincing level of realism. It is often necessary to provide LOD techniques in order to deliver real-time computer graphics and animations. Therefore, *mesh simplification* is adopted for efficient rendering and transmission. The most common use of mesh simplification is to generate *multiresolution models* or various *levels of detail* (LOD). For example, closer objects are rendered with a higher LOD, and distant objects with a lower LOD. Thanks to LOD management, many applications such as CAD visualization can accelerate rendering and increase interactivity. A survey on mesh simplification can be found in Ref. [36].

The most popular polygon-reduction technique is an *edge collapse* or simply *ecol* (more generally, vertex merging or vertex pair contraction) where two vertices are collapsed into a single one. The issues in *ecol* include which vertices to merge in what order, where to place the resulting vertex, etc. *Vertex split* or simply *vsplit* is the inverse operation of *ecol*. These operations are illustrated in Fig. 1(a) and a sequence of operations is illustrated on a sample model given in Fig. 1(b).

Hoppe proposed the *progressive mesh* (PM) [38], which consists of a coarse base mesh (created by a sequence of *ecol operations*) and a sequence of *vsplit operations*. Applying a subset of *vsplit operations* to the base mesh creates an intermediate simplification. The *vsplit* and *ecol* operations are known to be fast enough to apply at runtime, therefore supporting dynamic simplification.

### 3 Role-Based Viewing

In the context of 3D design, a *model M* is a description of an artifact, usually an individual part or assembly, in the form of a solid model. A true collaborative engineering environment enables multiple engineers to simultaneously work with *M*. The engineers (designers, process engineers, etc) correspond to a set of *actors*  $A = \{a_0, a_1, \dots, a_n\}$  each of which has associated with it a set of *roles*. Roles,  $R = \{r_0, r_1, \dots, r_m\}$ , define access and interaction rights for the actors. For example, actor  $a_3$  might have associated with it roles,  $r_{20}$ ,  $r_{23}$ , and  $r_{75}$ —this entitles  $a_3$  to view (and perhaps change) portions of *M* associated with these roles. Portions of *M* not associated with these roles, however, might be “off limits” to actor  $a_3$ . This section will build on the results of Ref. [2], where *Role-based Viewing* was developed in the context of distributed collaborative CAD, by introducing role hierarchies and their relation to multiresolution surfaces.

We formulate the problem of *role-based viewing* in the following subsections by developing:

- Actor-role framework: a general RBAC framework for describing actors and roles within a collaborative-distributed design environment. This framework uses a hierarchical graph to capture role-role relationships and create a relation between actors and roles.
- Model-role framework: an associative mapping from roles to topological regions on models. These regions

capture the *security features*,  $F$ , of a 3D model—relating how a point, patch, part, or assembly can be viewed by actors with given roles.

- Hierarchical role-based viewing: an algorithm to generate a *role-based view* given an actor  $a$ , his/her set of roles, the role hierarchy (RH), a model  $M$ , and its set of security features. A *role-based view* is a tailored 3D model which is customized for actor  $a$  based on the roles defining  $a$ 's access permissions on the model. In this way, the role-based view model does not compromise sensitive model information which  $a$  is not allowed to see (or see in detail). This is accomplished using a mesh simplification technique to generate the *role-based view*.

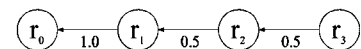
**3.1 Actor-Role Security Framework.** Our security framework is based on an adaptation of role-based access control, as developed in the information assurance and security literature [39], to the collaborative design problem. We focus on the relation between actors, their roles and the solid model geometry. This is in contrast to other work on access control in collaborative CAD which has focused mainly on database synchronization/transaction issues [40].

Representing actors and roles: We define a hierarchical RBAC framework where:

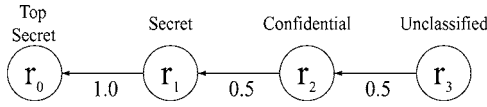
- Entities include a set of actors,  $A = \{a_0, a_1, \dots, a_n\}$  and a set of roles  $r = \{r_0, r_1, \dots, r_m\}$ ;
- Actor-role assignment, AR, is a relation (possibly many-to-many) of actors to roles:  $AR \subseteq A \times R$ ;
- Role hierarchy, RH, captures the relationships among the roles. For example, the permissions entailed by role  $r_{75}$  might be a superset of those entailed by. Hence, the role hierarchy is a *weighted*, directed acyclic graph (DAG),  $RH = (R, H)$ , where  $H \subseteq R \times R$  is the hierarchical set of relationships (edges) among the roles in  $R$ . This creates a partial order on  $R$ , hence (in the example above) if  $\langle r_{23}, r_{75} \rangle \in H$  then  $r_{23} < r_{75}$ . The weight of each edge in  $H$  is given by the real-valued function  $w: H \rightarrow [0, 1]$ .

An example of this RBAC framework is given in Fig. 2. For the remainder of this paper, we focus on *read/viewing permissions* granted by a given set of roles. Rather than “all or nothing” read permissions, our objective is to assign a “degree of visibility” to features of a model based on an actor’s roles. Using this formulation, we show how one can implement a Bell-La Padula-based [6] security model for collaborative viewing of CAD data.

Visibility	$a_0$	$a_1$	$a_2$	$a_3$
$r_0$	•			
$r_1$		•		
$r_2$			•	
$r_3$				•



**Fig. 2** Example actor-role (AR) and role hierarchy (RH) assignments



**Fig. 3 An example weighted role hierarchy with associated labels**

*Example:* Using the simple actor-role assignment matrix and role hierarchy from Fig. 2, we can compute the degree of visibility to each actor for a model assigned to a specific role. To implement the Bell-La Padula [6] model, we need to compute visibility in such a way as to guarantee that the role (e.g., security clearance) of someone receiving a piece of information must be at least as high as the role assigned to the information itself. In this way, a CAD model classified as “Secret” can only be viewed by those with a “Top-Secret” or “Secret” classification, but not viewed by someone with only a “Confidential” level of access. Figure 3 illustrates this example.

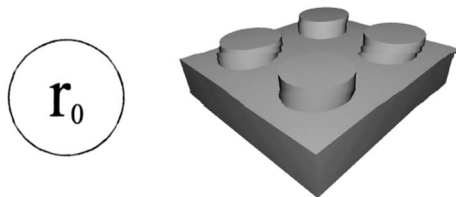
**3.2 Model-Role Security Framework.** Let  $M$  be a solid model of an artifact (part, assembly, etc.) and let  $b(M)$  represent the boundary of  $M$ . In this context, the **Model-Role Assignment**,  $MR$ , is a relation (possibly many-to-many) assigning roles to points on the surface of the model:  $MR \subseteq b(M) \times R$ , where each point on  $b(M)$  has at least one role (i.e.,  $\forall p \in b(M), \exists r \in R$  such that  $\langle p, r \rangle \in MR$ ). In this way, each point on the surface of the solid model  $M$  has associated with it some set of access rights dependent on the roles associated with it.

In practice, it is impractical to assign roles point-by-point to the  $b(M)$ . Hence, we define a set of *security features*,  $F = \{f_0, f_1, \dots, f_k\}$ , where each  $f_i$  is a topologically connected patch on  $b(M)$  and  $\cup F = b(M)$ ; and each  $f_i$  has a common set of role assignments. Therefore, the Model-Role assignment can be simplified to be the relation associating security features with access roles:  $MR \subseteq F \times R$ .

*Example:* Let  $M$  be a solid model; let,  $F = \{f_1\}$ , where  $f_1 = b(M)$  (i.e., the entire boundary is one security feature). If  $MR = \{\langle f_1, r_0 \rangle\}$  (where  $r_0$  is from the previous example in Fig. 3), then we can see the resultant model for  $r_0$  depicted in Fig. 4.

**3.3 Hierarchical Role-Based Viewing.** The issue now is that, for a given actor  $a$ , what portions of the model  $M$  that he/she can see will depend on their associated roles and the security features of the model. Depending on their permissions, a new model,  $M'$ , must be generated from  $M$  such that the security features are not shown or obfuscated based on the actor’s roles. If their roles give them permission to see certain features (i.e., mating features), then the resulting model includes the features with the same fidelity [41] as in  $M$ ; if not, the features must be obfuscated in such a way as to hide from  $a$  what  $a$  does not have the right to see. Hence, the *role-based view* generation problem can be stated:

*Problem:* Given a set of roles and their relationships ( $R$  and  $RH$ ); a solid model and its security features ( $M$ ,  $F$ , and  $MR$ ); and an actor ( $a$  and  $AR$ ), determine the appropriate view  $M'$  of model



**Fig. 4 An example part with one security feature where  $b(M)$  is assigned to  $r_0$**

$M$  for actor  $a$ .

We propose a solution based on the use of multiresolution meshes, as follows:

- (1) Convert solid model  $M$  to a high-fidelity mesh representation;
- (2) Based on  $F$ , determine which facets belong to each security feature,  $f$ ;
- (3) For each security feature  $f$ , do:
  - (a) If the intersection of actor  $a$ ’s roles and  $f$ ’s roles is nonempty, then add the facets associated with  $f$  to  $M'$ ;
  - (b) If actor  $a$ ’s roles do not intersect the roles of  $f$ , determine (using RH) how much of  $f$  they are allowed to see and create a set of modified facets to represent  $f$  for inclusion in  $M'$ .
- (4) Clean up the resulting  $M'$  so that boundaries of the  $f_i$ ’s are topologically valid.
- (5) Return  $M'$ .

There are three research problems we address:

- (1) How does the role-hierarchy RH relate to the degree of visibility?

We show how the weighted DAG that comprises RH can be used to implement a number of useful security policies by making the model quality a function of the path-cost among roles in RH.

- (2) How to modify the facets for each  $f_i$  based on RH?

Our approach is to use a security policy (based on Bell-La Padula) associated with the role hierarchy RH to determine how to modify the model. In some cases, policy will dictate degradation of the model fidelity; in other cases, the security features may be completely deleted or replaced with a simple convex hull or bounding box as in Ref. [2]. To accomplish this, we employ multiresolution meshes: model fidelity will be preserved to the degree the actor’s rights allow it. The result is a mesh appropriate for viewing by the actor  $a$ .

- (3) How to ensure that the resulting regions form a topologically valid model?

Deforming the model feature by feature may result in topological regions of facets in  $M'$  that are misaligned or aesthetically unpleasing. Cracks and occlusion can be avoided by preserving the boundary edges during simplification.

*Example:* This example shows a model  $M$  whose surface is described by one security feature  $f_0$ . Given the role-hierarchy from Fig. 3, and four actors,  $a_0$ ,  $a_1$ ,  $a_2$ , and  $a_3$  with their AR shown in Fig. 2. Figure 5 shows the four different views of model  $M$  they each see. Given the AR, RH, and MR assignments, we can derive the direct *actor*  $\times$  *feature* mappings. Figure 6 gives the direct mappings specified implicitly by the AR, RH, and MR given in Figs. 2(a), 2(b), and 5 respectively. The two MR assignments that are not shown are  $f_1 \in r_1$  and  $f_2 \in r_2$ . It is important to note that, similar to inheritance found in most object-oriented programming languages,  $a_0$  cannot see  $f_1$  or  $f_2$  even though it is the base role for subroles  $r_1$ ,  $r_2$ , and  $r_3$ . Hence an inheritance relation allows a child to inherit the permissions of the parent, but nothing is implied in the other direction.

## 4 Technical Approach

We combine techniques from solid modeling and computer graphics to provide a secure collaborative environment which supports real-time design. In this section we describe how to modify and configure Hierarchical RBAC to support our multiresolution security model. We describe the problems, algorithms employed, and final considerations.

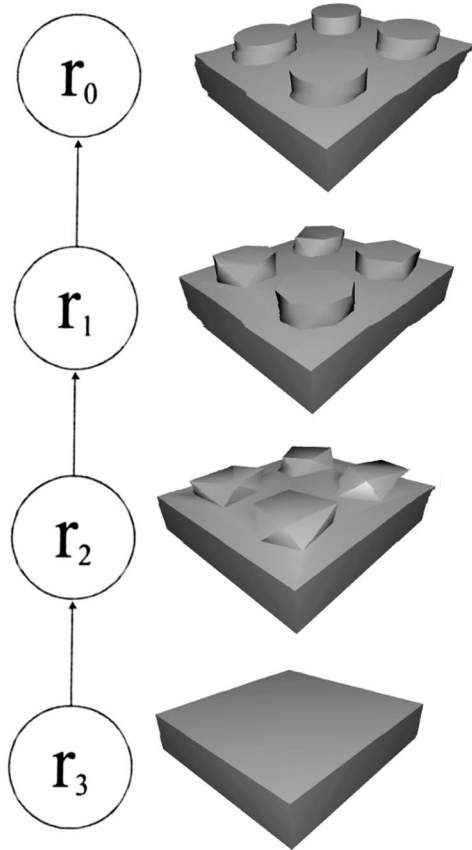


Fig. 5 An example part with one security feature ( $f_0$ ) consisting of  $b(M)$  assigned to  $r_0$ , a set of actors assigned to roles, and their corresponding set of secure models

**4.1 Hierarchical RBAC Policy.** Since RBAC is a means of articulating policy rather than a policy by itself, an actual policy is necessary. We wish to adopt a policy similar to the classical MAC model [6]. This is defined in terms of the following axioms using  $\lambda$  to return the security level of either an actor or a feature:

- (1) Simple security property: Actor  $a$  can read feature  $f$  iff  $\lambda(a) \geq \lambda(f)$ . This is also known as the read-down property.
- (2) Liberal \*-property: Actor  $a$  can write feature  $f$  iff  $\lambda(a) \leq \lambda(f)$ . This is also known as the write-up property.

There are many variations of the \*-property, but we will focus on the simple security property which essentially states that the clearance of a person receiving a piece of information must be at least as high as the classification of the object. Details on a formal construction of MAC in RBAC have been presented by Osborn [5].

Hierarchical RBAC is a natural means for structuring roles that

	$f_0$	$f_1$	$f_2$
$a_0$	1.0	$n/a$	$n/a$
$a_1$	1.0	$n/a$	$n/a$
$a_2$	0.5	1.0	$n/a$
$a_3$	0.25	0.5	1.0

Fig. 6 The direct permission mappings derived from the AR, RH, and MR relations given in Figs. 2(a), 2(b), and 5, respectively

reflect an organization's lines of authority and responsibility [39]. The main distinction between our approach and the generic RBAC frameworks found in literature, is that *we also allow permissions to be modified through the role hierarchy*. Typically *permissions* (i.e., an object and a permissible operation) are associated with every combination of  $object \times role$ . Since our read permissions are specified by a degree of visibility value, an inheritance relation can further refine this value. An inheritance relation is a binary relation (*parent, child*), where the child inherits permissions from the parent based upon a multiplicative weight  $w$ . For instance:  $w=1.0$  preserves the parents permissions exactly, while  $w=0.5$  will reduce the degree of visibility by half for all inherited objects. By transitivity, this weighted factor applies to all inherited objects specified in the role hierarchy.

Intuitively, it might appear that we are breaking the simple security property by allowing some actors to view objects that they normally would not be able to see. This is not the case, and instead should be viewed as transforming one object into a new object that is permissible. Hence, our model still adheres to the simple security property.

Given an actor ( $a$ ) and a feature ( $f$ ), the test to determine if  $a$  has permissions on  $f$  is equivalent to computing graph reachability among all possible pairs of roles assigned to both  $a$  and  $f$ . We will use  $R_a$  to denote the set of roles assigned to  $a$ , and  $R_f$  for the set of roles assigned to  $f$ . If any role in  $R_a$  is reachable from any other role in  $R_f$  (i.e., there exists a path), then the product of all weights along the path yields the degree of visibility for that path. We will use a reachability function to return the set of all roles reachable from a given role. Several paths are possible, hence the resultant degree of visibility for  $a$  will be chosen as the maximum. We denote the function that returns the maximum degree of visibility for  $a$  on  $f$  as  $\alpha(a, f)$ . The result of this function can be computed once, stored, and reused until an existing role assignment (AR or MR) is modified. The degree of visibility is then used as a parameter to another function which degrades the fidelity of a feature depending on an actors permissions.

**4.2 Generation of Multilevel Security Models.** For part/component/assemblies with regions that need to be secured, multiresolution techniques are employed to provide various levels of detail. Although the original (highest) resolution version of a model might be a breach for some actors, lower resolution LODs will be sufficiently secure to transmit to those actors. In addition to purely geometric multiresolution techniques, Shyamsundar and Gadh have developed a framework for representing different levels of detail for geometric feature data [31,42]. Our security model could be used in conjunction with this feature LOD representation, but an automatic simplification algorithm needs to be developed.

Mesh simplification techniques include either vertex decimation, vertex clustering, or edge contraction. Choosing a specific simplification technique among the breadth of candidates is application dependent. To address the demands of an interactive collaborative design environment, we outline several issues which are critical for simplification:

- (1) *Speed*: As the number of component/assemblies in a session increases, the simplification becomes the bottleneck. We need an algorithm capable of drastic simplification in the least amount of time.
- (2) *Continuous*: A continuous spectrum of detail is necessary so an appropriate model can be selected at runtime. We do not wish to store all possible LODs within the model repository due to space constraints, but parameters of the simplification hierarchy can be precomputed and stored.
- (3) *Boundary preserving*: The boundary of objects should be preserved in order to distinguish objects from one another. Inadvertent occlusion and cracks may result if we relieve this constraint.
- (4) *View-independence*: The viewer receives 3D model infor-

mation therefore the simplification should also support this. This allows clients to perform trivial viewing operations without the incremental expense in network bandwidth.

- (5) *Fidelity*: The quality or similarity of a modified model with respect to its nominal form is regarded as visual fidelity. This is a characteristic that ultimately can only be assessed by humans, but several studies have focused on this topic [41].

Given our requirements, Quadric Error Metrics [43] (QEM) is an obvious candidate. QEM provides drastic simplification, capable of progressivity, in a relatively small amount of time. This algorithm also produces a result that is realistic and recognizable as a simplified variant of the original model. One issue is the algorithms dependence upon a threshold value. In the rare case that the threshold value is as large as the model itself, then the algorithm runs in  $O(n^2)$ . An alternative approach is to compute an optimal threshold adaptively [44].

We have proposed using an automatic simplification technique to degrade the fidelity of a model enough to satisfy the access-control requirements of a collaborative design session. An automatic technique cannot be proven to sufficiently degrade the model enough to be secure in all environments. The process can be supplemented by adopting a form of user-guided simplification [45,46]. User-guided simplification is a means of supervising the simplification by editing the order of *ecol* performed during simplification, selecting regions where more or less simplification is necessary, or directly manipulating the vertex hierarchy. A side effect is that these simplification parameters need to be stored with the model, since these cannot be automatically derived.

QEM simplification can be configured to either maintain or modify the topological genus of a model. In a multiuser CAD server, progressive meshes (PM) [38] can be useful for the transmission of CAD models. If PM is used, and if the removal of holes yields a more secure version of a particular model, then genus-reduction techniques must be employed since standard PM is not compatible with topology-modifying simplification.

Cracks and occlusion must be avoided for continuous and adjacent regions of a part that are simplified independently. If a single part is partitioned into two or more regions, and each region has a different model-role assignment, then the regions will be simplified at different levels of detail. If boundary edges of the mesh are not preserved, then possibly cracks and self-occlusion will result.

## 5 Realization of Approach

The FACADE system is a multipurpose CAD framework that supports numerous modes of functionality implemented as modules. Its most basic component is a 3D model viewer that supports standard camera navigation operations and the ability to view models using different shading algorithms or as a wireframe. FACADE's design allows instances of the system to be compiled with or without a particular module.

The first module is a light design module which enables several basic tasks such as: selection of a part, component, assembly, or other selectable entity; applying affine transformations on a part; adding an alpha channel to a part for transparency; decomposing a part into multiple parts; specifying a set of parts as an assembly; manipulating control points on parametric surfaces (e.g., bezier patches, splines, and NURBS). This subsystem is a prerequisite for most of the remaining modules described below.

The security authoring module provides the interface which allows a designer to assign *role-based viewing* parameters to a part, component, assembly, or semantic features that can be saved and later reloaded. The designing stage allows a designer to assign a {label, permission}-tuple to parts, assemblies, or individual facets. The normalized permissions [0.0–1.0] were used to indicate a percentage of the features to be suppressed from the original model. In situations where the result is not sufficiently secure, a super-

vised technique, such as user-guided simplification can be used.

When a designer requests a model, the user must first declare their identity so all direct role associations can be retrieved and implied associations, from RH, can be derived. Based upon the roles associated with a designer and the model features, a *role-based view* is generated. We used a single administrative account to modify permissions in the model repository. There are numerous administrative configurations which have been presented by Sandhu [47]. The goals and constraints of the collaboration will dictate how comprehensive the role administration requirements should be.

We have implemented our own topology-preserving QEM-based simplification algorithm. For the experiments in this paper, we chose to collapse only vertex pairs which are connected by an edge. The simplification algorithm is passed each tessellated and triangulated part, or connected region of a part with an equivalent set of tuples (label, permission). Since these regions are disjoint, they can be simplified and transmitted in parallel. The running time Data was collected in Ref. [43] on an SGI Indigo2 with a 195 MHz R10000 processor and 128 Mbytes of memory of QEM on a model with nearly 70,000 facets simplified to 10 facets was done in approximately 15 s, and only 12 s if the error heap was precomputed [43]. We can precompute this heap structure on the server when a model is saved for faster transmission the next time it is loaded.

The last module in FACADE enables the network client interface that can talk to a FACADE server. The server works in conjunction with the security module to provide *role-based views* to clients which do not have permissions to manipulate or view a model, or its semantic features, at full resolution. The server maintains consistency throughout all connected clients by sending rejection messages to clients when a design operation they have performed conflicts with the operation of another client. The server supports both "thin" and "fat" FACADE network clients and their corresponding protocols. The thin clients communicate using the Remote Frame Buffer (RFB) protocol [48]. The fat clients communicate with an unpublished text-based protocol which sends only design transformation information after the initial model is sent.

The FACADE framework has been designed for maximum portability across all platforms. It has been tested and simultaneously developed in Solaris/SunOS (Sun CC/GNU g++), Linux (GNU g++), and Windows 2000 (Microsoft Visual C++) operating systems. It is implemented in C++ using OpenGL as the graphics rendering library. An OpenGL *canvas* can be displayed using either GLUT or Java via the JNI interface. The network socket libraries use BSD-style sockets under Unix-based derivatives and Winsock2 under Windows. The multithreaded code uses POSIX threads (pthreads) under Unix-based derivatives and Windows Threads under Windows.

## 6 Example: Computer Mouse Assembly

We show how role hierarchies can be used to instantiate a multilevel information security model on an electromechanical assembly. Unlike previous work in role-based viewing [2], this example demonstrates how the weighted role hierarchy affects the collaboration space. We present this as a more flexible and practical approach since role hierarchies naturally reflect an organization's line of authority and responsibility.

In the following example scenario, six actors are granted permission to view and modify a mouse assembly at some level of abstraction. Each actor is assigned a label and assigned a role from the role hierarchy. The set of actors is partitioned into three groups based on the nature of the design work: electrical, mechanical, and ergonomic. Individual parts, regions of parts, or other feature information are assigned labels and grouped into one of the hierarchies. These labels and descriptions are given in Fig. 7.

Label	Actor	Hierarchy	Label	Feature	Hierarchy
$a_0$	Supervisor	All	$f_0$	Cable	Electrical
$a_1$	Electrical Engineer	Electrical	$f_1$	Circuit Board	Electrical
$a_2$	Electrical Observer	Electrical	$f_2$	Cable/Board Interface	Electrical
$a_3$	Mechanical Engineer	Mechanical	$f_3$	Ball	Mechanical
$a_4$	Mechanical Observer	Mechanical	$f_4$	Ball Housing	Mechanical
$a_5$	Ergonomics Engineer	Ergonomic	$f_5$	Lower Casing	Ergonomic
			$f_6$	Buttons	Ergonomic

Fig. 7 Actor×feature labels and descriptions for the mouse assembly example

Each of the lead engineers will be given full permissions to their respective subsystems. They are each given a subordinate, or observer, who is in training for this design. The subroles created for this purpose will be called observer roles. The engineers of one particular hierarchy will also need some level of viewing permissions to the other hierarchies, especially at the interface features. These roles will be called *interface roles*. The roles for the electrical, mechanical, and ergonomic hierarchies are given as  $r_e$ ,  $r_m$ , and  $r_s$ , respectively. We'll use a second subscript to denote the role's position in the hierarchy where 0 is labeled as the lead roles, 1 is labeled as the observer roles, and 2 is for the interface roles. This actor-role assignment matrix is given in Fig. 8(a). The observer and interface roles are given less viewing privileges than the lead roles. The observer roles are given half the degree of visibility as the lead roles, and the interface roles are given half the degree of visibility as the as the observer roles. This weighted hierarchy is depicted in Fig. 8(b). Figure 8(c) gives the complete set of model-role assignments for the mouse assembly. In this example, one clear advantage of the role hierarchy is that model-role assignments exist for only lead engineers and the subordinate roles inherit those permissions. Using  $a(a,f)$ , Fig. 8(d) gives the degree of visibility values computed for every *actor*×*feature*. These values are implicit in the AR, RH, and MR structure.

The supervisor ( $a_s$ ) has unrestricted access to all features of the assembly. The supervisor's role-based view is given in Fig. 9(a). Figure 9(b) shows the role-based view for the electrical engineer ( $a_{e,0}$ ) which shows the electrical features in full resolution, the mechanical features in a lower resolution, and the exterior features in an even lower resolution. Figure 9(c) gives the role-based view for the mechanical engineer where the mechanical features are displayed in full resolution, the electrical features in a lower resolution, and the exterior features in even lower detail. Figure 9(d) depicts the ergonomics engineer's ( $r_{s,0}$ ) role-based view which depicts the interior and exterior in full resolution, but the remaining features are displayed in a low resolution. By using role-based

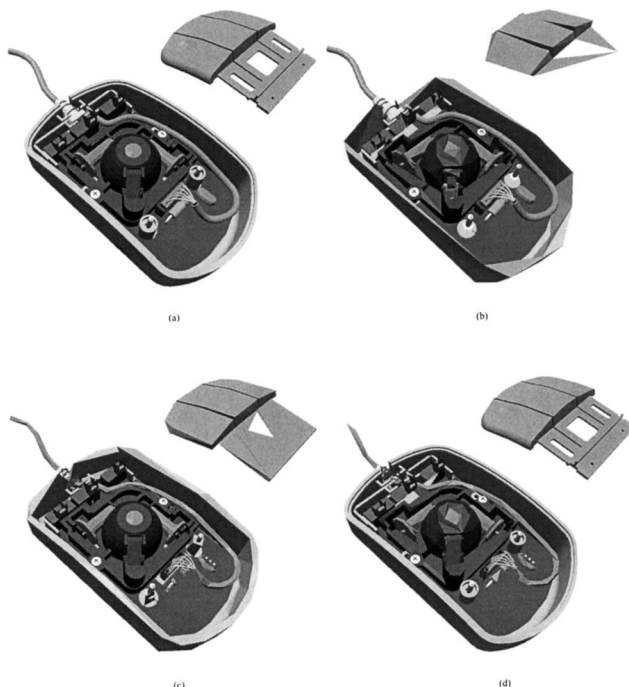


Fig. 9 Role-based views for the mouse assembly example. (a) The supervisor's ( $a_0$ ) full resolution view. (b) The electrical engineer's ( $a_1$ ) role-based view. The electrical features are displayed in full resolution, the mechanical features in a lower resolution, and the exterior features in an even lower resolution. (c) The mechanical engineer's ( $a_3$ ) role-based view. Mechanical features are displayed in full resolution, the electrical features in a lower resolution, and the exterior features in even lower detail. (d) The ergonomic engineer's ( $a_5$ ) role-based view. This depicts the interior and exterior in full resolution, but the remaining features are displayed in a low resolution.

views, designers need not be concerned with unnecessary design details and the protection of sensitive intellectual property can be maintained.

## 7 Conclusions and Future Work

This paper developed a new technique, *Hierarchical Role-Based Viewing*, for multilevel information security in collaborative 3D assembly design. Role hierarchies naturally reflect an or-

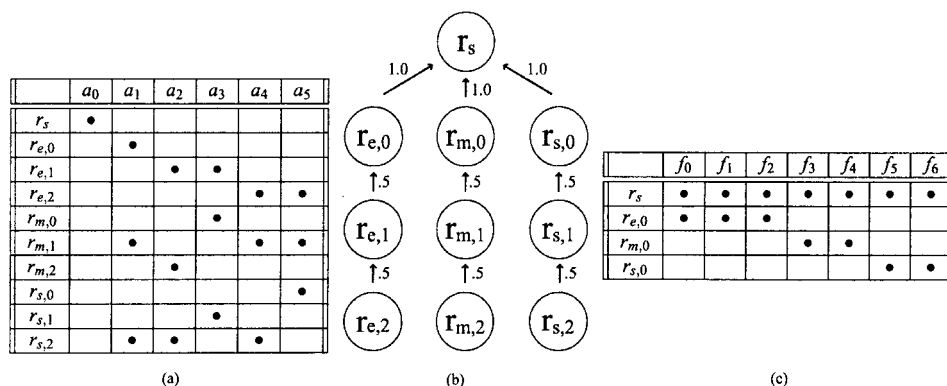


Fig. 8 All security associations and derived mappings for the mouse assembly example: (a) Actor-Role (AR) assignment matrix; (b) Weighted DAG representing the Role Hierarchy (RH). The  $r_e$  hierarchy is assigned the electrical subsystem, the  $r_m$  hierarchy is assigned the mechanical subsystem, and the  $r_s$  hierarchy is assigned the shell; (c) Model-Role (MR) assignment matrix.

ganization's line of authority and responsibility. By incorporating security with collaborative design, the costs and risks incurred by multiorganizational collaboration can be reduced. Aside from digital 3D watermarking, research on how to provide security issues to distributed design, working in collaborative graphical environments, remains a novel and relatively unexplored area. The authors believe that this work is the first of its kind to bring multilevel security to geometric data in the field of computer-aided design and collaborative engineering.

Immediate future work involves using multiresolution techniques directly on the native surface types and examining network configurations to reduce aggregate bandwidth. We are currently extending these techniques to handle B-spline surfaces directly. The motivation for handling these surfaces is to demonstrate that for certain geometry, multiresolution surface techniques will provide a more intuitive simplification result. Crack prevention, permissions on patch boundaries when adjacent patches have different roles, and other issues will need to be addressed. We would also like to give a demonstration of the model on geometric, as well as, semantic, feature data.

Our environment has been extended to support synchronous multiuser collaborative CAD. Optimal network configurations can be constructed and "grouping" of the mesh hierarchy can be performed for actors and assigned similar roles. We can take advantage of continuous LOD over a network using a progressive technique, such as Progressive Meshes [38]. This results in computing only one mesh hierarchy for an entire set of actors. For further optimization, multicast networks could be used to properly aggregate bandwidth when actors have similar privileges.

## Acknowledgment

This work was supported in part by the National Science Foundation (NSF) Knowledge and Distributed Intelligence in the Information Age (KDI) Initiative Grant No. CISE/IIS-9873005; CA-REER Award No. CISE/IIS-9733545; Office of Naval Research (ONR) Grant No. N00014-01-1-0618; and the Science and Technology Program of Gyeonggi province. This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Science and Technology of Korea. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or the other supporting government and corporate organizations.

## References

- [1] Callahan, S., and Heisserman, J., 1996, "A Product Representation to Support Process Automation," in *Product Modeling for Computer Integrated Design and Manufacture*, M. Pratt, R. Sriram, and M. Wozny, eds., Chapman and Hall, New York, pp. 285–295.
- [2] Cera, C. D., Kim, T., Han, J., and Regli, W. C., 2004, "Role-Based Viewing Envelopes for Information Protection in Collaborative Modeling," *Comput.-Aided Des.*, **36**(9), pp. 873–886.
- [3] Praun, E., Hoppe, H., and Finkelstein, A., 1999, "Robust Mesh Watermarking," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley, Reading, pp. 49–56.
- [4] Lampson, B., 1971, "Protection," in *Proceedings of the 5th annual Princeton conference on information sciences and systems*, pp. 437–443.
- [5] Osborn, S. L., Sandhu, R. S., and Munawer, Q., 2000, "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies," *Entrepreneurship Theory Pract.*, **3**(2), pp. 85–106.
- [6] Bell, D., and LaPadula, L., 1973, "Secure Computing Systems: Mathematical Foundation and Model," *MITRE Report No. (MTR 2547) v2*.
- [7] Sandhu, R. S., and Samarati, P., 1994, "Access Control: Principles and Practice," *IEEE Commun. Mag.*, **32**(9), pp. 40–48.
- [8] Jajodia, S., and Sandhu, R., 1991, "Toward a Multilevel Secure Relational Data Model," in *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data*, ACM, New York, pp. 50–59.
- [9] Sandhu, R., and Chen, F., 1998, "The Multilevel Relational (MLR) Data Model," *ACM Trans. On Inf. Sys.*, **1**(1), pp. 93–132.
- [10] Fisher, H., 2002, "Multicast Issues for Collaborative Virtual Environments," *IEEE Comput. Graphics Appl.*, **22**(5), pp. 68–75.
- [11] Frecon, E., and Stenius, M., 1998, "DIVE: A Scalable Network Architecture for Distributed Virtual Environments," *Distributed Systems Engineering Journal* (Special Issue on Distributed Virtual Environments), **5**(3), pp. 91–100.
- [12] Greenhalgh, C. M., and Benford, S. D., 1999, "Supporting Rich and Dynamic Communication in Large Scale Collaborative Virtual Environments," *Presence*, **8**(1), pp. 14–35.
- [13] Dias, J. M. S., Galli, R., Almeida, A. C., Bello, C. A. C., and Rebordao, J. M., 1997, "mWorld: A Multiuser 3D Virtual Environment," *IEEE Comput. Graphics Appl.*, **17**(2), pp. 55–65.
- [14] Barrus, J. W., and Waters, R. C., 1996, "Locales: Supporting Large Multiuser Virtual Environments," *IEEE Comput. Graphics Appl.*, **16**(6), pp. 50–57.
- [15] Funkhouser, T. A., 1995, "RING: A Client-Server System for Multi-User Virtual Environments," in *Symposium on Interactive 3D Graphics*, pp. 85–92, 209.
- [16] Macedonia, M., Zyda, M., Pratt, D., Barham, P., and Zeswitz, S., 1994, "NPS-NET: A Network Software Architecture for Large-Scale Virtual Environments," *Presence*, **3**(4), pp. 265–287.
- [17] Eriksson, H., 1994, "MBONE: The Multicast Backbone," *Commun. ACM*, **37**(8), pp. 54–60.
- [18] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., and Hart, P., 1999, "VADE: a Virtual Assembly Design Environment," *IEEE Comput. Graphics Appl.*, **19**(6), pp. 44–50.
- [19] Conner, B., Cutts, M., Fish, R., Fuchs, H., Holden, L., Jacobs, M., Loss, B., Markosian, L., Riesenfeld, R., and Turk, G., 1997, "An Immersive Tool for Wide-Area Collaborative Design," in *TeamCAD, the First Graphics Visualization, and Usability (GVU) Workshop on Collaborative Design*, pp. 139–143.
- [20] Bajaj, C., and Cutchin, S., 1997, "Web Based Collaboration-Aware Synthetic Environments," in *TeamCAD, the First Graphics Visualization, and Usability (GVU) Workshop on Collaborative Design*, pp. 143–150.
- [21] Cutkosky, M. R., Tenenbaum, J. M., and Glicksman, J., 1996, "Madefast: Collaborative Engineering Over the Internet," *Commun. ACM*, **39**(9), pp. 78–87.
- [22] Qun, L., Tay, F. E. H., Cheng, T. K., Kuen, K. K., and Feng, Z. Y., 2000, *CyberCAD: Internet Distributed Interactive Collaborative Design. Demonstrated in the International Grid Application Research Demonstrations, INET 2000. The Internet Global Summit. Yokohama, Japan, July*.
- [23] Wu, D., and Sarma, R., 2004, "The Incremental Editing of Faceted Models in an Integrated Design Environment," *Comput.-Aided Des.*, **36**(9), pp. 821–833.
- [24] Qiu, Z. M., Wong, Y. S., Fuh, J. Y. H., Chen, Y. P., Zhou, Z. D., Li, W. D., and Lu, Y. Q., 2004, "Geometric Model Simplification for Distributed CAD," *Comput.-Aided Des.*, **36**(9), pp. 809–819.
- [25] El-Sana, J., Evans, F., Kalaiyah, A., Varshney, A., Skiena, S., and Azanli, E., 2000, "Efficiently Computing and Updating Triangle Strips for Real-Time Rendering," *Comput.-Aided Des.*, **32**(13), pp. 753–772.
- [26] De Floriani, L., Magillo, P., Morando, F., and Puppo, E., 2000, "Dynamic View-Dependent Multiresolution on a Client Server Architecture," *Comput.-Aided Des.*, **32**(13), pp. 805–823.
- [27] Luebke, D., 1999, "Robust View-Dependent Simplification for Very Large-Scale CAD Visualization," *Technical Report No. CS-99-33, University of Virginia, Department of Computer Science, Charlottesville, Virginia*.
- [28] Li, W. D., Ong, S. K., Fuh, J. Y. H., Wong, Y. S., Lu, Y. Q., and Nee, A. Y. C., 2004, "Feature-Based Design in a Distributed and Collaborative Environment," *Comput.-Aided Des.*, **36**(9), pp. 775–797.
- [29] Bidarra, R., van den Berg, E., and Bronsvort, W., 2002, "A Collaborative Feature Modeling System," *J. Comput. Inf. Sci. Eng.*, **2**(3), pp. 192–198.
- [30] Bidarra, R., Kranendonk, N., Noort, A., and Bronsvort, W., 2002, "A Collaborative Framework for Integrated Part and Assembly Modeling," *J. Comput. Inf. Sci. Eng.*, **2**(4), pp. 256–264.
- [31] Shyamsundar, N., and Gadh, R., 2001, "Internet-Based Collaborative Product Design with Assembly Features and Virtual Design Spaces," *Comput.-Aided Des.*, **33**(9), pp. 637–651.
- [32] Wu, D., and Sarma, R., 2001, "Dynamic Segmentation and Incremental Editing of Boundary Representations in a Collaborative Design Environment," in *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, ACM, New York, pp. 289–300.
- [33] Lee, J., Kim, H., Han, S., and Park, S., 1999, "Network-Centric Feature-Based Modeling," in *Proceedings of the Pacific Graphics Conference on Computer Graphics and Applications*, pp. 280–289.
- [34] Piegl, L. A., and Richard, A. M., 1995, "Tessellating Trimmed Nurbs Surfaces," *Comput.-Aided Des.*, **27**(1), pp. 16–26.
- [35] Sheng, X., and Hirsch, B. E., 1992, "Triangulation of Trimmed Surfaces in Parametric Space," *Comput.-Aided Des.*, **24**(8), pp. 437–444.
- [36] Luebke, D. P., 2001, "A Developer's Survey of Polygonal Simplification Algorithms," *IEEE Comput. Graphics Appl.*, **21**(3), pp. 24–35.
- [37] National Design Repository, 2003, National Design Repository, <http://www.designrepository.org>
- [38] Hoppe, H., 1996, "Progressive Meshes," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, NY, pp. 99–108.
- [39] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E., 1996, "Role-Based Access Control Models," *IEEE Comput. Graphics Appl.*, **29**(2), pp. 38–47.
- [40] Bancilhon, F., Kim, W., and Korth, H. F., 1985, "A Model of CAD Transactions," in *VLDB'85 Proceedings of the 11th International Conference on Very Large Data Bases*, A. Pirotte and Y. Vassiliou, eds., Morgan Kaufmann, pp. 25–33.
- [41] Watson, B., Friedman, A., and McGaffey, A., 2001, "Measuring and Predict-



- ing Visual Fidelity,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, NY, pp. 213–220.
- [42] Shyamsundar, N., and Gadh, R., 2002, “Collaborative Virtual Prototyping of Product Assemblies Over the Internet,” *Comput.-Aided Des.*, **34**(10), pp. 755–768.
- [43] Garland, M., and Heckbert, P. S., 1997, “Surface Simplification Using Quadric Error Metrics,” in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., pp. 209–216.
- [44] Erikson, C., and Manocha, D., 1999, “GAPS: General and Automatic Polygonal Simplification,” in *Symposium on Interactive 3D Graphics*, pp. 79–88.
- [45] Li, G., and Watson, B., 2001, “Semiautomatic Simplification,” in *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM, NY, pp. 43–48.
- [46] Kho, Y., and Garland, M., 2003, “User-Guided Simplification,” in *Proceedings of the 2003 Symposium on Interactive 3D Graphics*, ACM, NY, pp. 123–126.
- [47] Sandhu, R., Bhamidipati, V., and Munawar, Q., 1999, “The ARBAC97 Model for Role-Based Administration of Roles,” *ACM Trans. On Inf. Sys. (TISSEC)*, **2**(1), pp. 105–135.
- [48] Richardson, T., and Wood, K. R., 1998, *The RFB Protocol Version 3.3. Protocol Specification*, July.