Yan Wang[1]
Assistant Professor
NSF Center for *e*-Design,
University of Central Florida,
4000 Central Florida Blvd.,
Orlando, FL 32816-2993
e-mail: wangyan@mail.ucf.edu

Pamela N. Ajoku
NSF Center for *e*-Design,
University of Pittsburgh,
1048 Benedum Hall,
Pittsburgh, PA 15261
e-mail: pne1@pitt.edu

José C. Brustoloni
Assistant Professor
Department of Computer Science,
University of Pittsburgh,
6111 Sennott Sq. Bldg.,
210 S. Bouquet St.,
Pittsburgh, PA 15260
e-mail: jcb@cs.pitt.edu

Bart O. Nnaji
William Kepler Whiteford Professor & Director
NSF Center for *e*-Design,
University of Pittsburgh,
1048 Benedum Hall,
Pittsburgh, PA 15261
e-mail: nnaji@engr.pitt.edu

Intellectual Property Protection in Collaborative Design through Lean Information Modeling and Sharing

Establishing efficient, effective, and trustworthy engineering collaboration while protecting intellectual property is vital to maintain organizational competence in today's global business environment. In this paper, a lean information modeling and sharing framework is described to support engineering data security management in a peer-to-peer collaborative environment. It allows for selective and interoperable data sharing with fine-grained access control at both the server and client sides, thus securing different levels of design information dissemination for intellectual property protection purposes. The considerations of time and value-adding activity with roles, policy delegation relation in a distributed context, and fine-grained control at data set level in the model are to adhere to the general least privilege principle in access control. Heterogeneous design data are exchanged selectively through an eXtensible Markup Language common interface, which provides a neutral format to enhance data interoperability and prevents reverse engineering. [DOI: 10.1115/1.2190235]

Keywords: internet-aided design and manufacturing, intellectual property protection, access control, digital rights management, data security, XML encryption, interoperability

1 Introduction

Global markets call for collaborative product development among designers, manufacturers, suppliers, vendors, and other stakeholders. The business pressures toward outsourcing allow most of the design work for complex products to be done across firms. For example, Ford estimates that its supply chain involves more than 800 suppliers, and Ford is substantively relying on these suppliers to participate in vehicle design [1]. The Defense Advanced Research Projects Agency (DARPA) estimates that the supply chain accounts for more than 50% of weapon system and major subsystem production costs [2].

Product design includes the whole spectrum of conceptualization, detailed design, analysis, simulation, and prototyping. Collaborative design is the process where multidisciplinary stakeholders participate in design decision-making and share product information across enterprise boundaries in an Internet-enabled distributed environment. Intellectual property (IP) protection thus is critical to a company when sharing data with its suppliers or customers. During collaboration, a manufacturer may share certain data with its supplier as design specifications. It may also share data with its customer for analysis and simulation purposes.

One of the most important items to protect in product development is design data such as certain parameter values, user defined features in feature models, special contours in surface models, inventive configuration in composite materials, and innovative assembly mechanisms, which may have IP value to protect. Particular attention to IP management is necessary in digital environments, in which perfect copies can easily be made at little cost. Establishing trustworthy engineering collaboration to protect IP is vital to maintain organizational competence.

Trust, confidentiality, and integrity issues involved in sharing data are immense. Two important security services needed for product data are *confidentiality* (of product design data in storage or in transit) and *access control* (read, write, delete privileges). There are new issues about design information modeling and communication in collaborative design. Current CAD data formats were designed for standalone systems. All information about components and assemblies has to be available locally in order to be processed. Transferring complete design information among design collaborators requires a large amount of data to be moved around. Some cryptographic communication protocols (e.g., SSL) provide good end-to-end security by securing the communication channel at the packet level and providing in-transit document confidentiality. However, data security should be ensured at either end of the communication link, especially at the client side.

Unlike conventional centralized access control models for files and resources, we present a fine-grained role-based access control for different data segments within one file in a distributed design environment in this paper. This model combines role-based and cryptographic access control to form a new mechanism for flexible data security management and selective information dissemination through interoperable data model in a collaborative environment. Heterogeneous data are shared through an eXtensible Markup Language (XML) common interface, which provides vendor-neutral solutions to enhance data interoperability.

There are many research issues related to collaborative design, for example, system architecture and infrastructure, data and system interoperability, conflict detection and resolution, version and concurrency control for collaborative modeling. However, in this

[1] Corresponding author.

Contributed by the Engineering Informatics (EIX) Committee of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received August 12, 2004; final manuscript received November 23, 2005. Assoc. Editor: S. Urban.

paper, we only focus on the protection issue in data sharing. In the rest of the paper, Sec. 2 describes the related work in data security, digital rights management, and intellectual property protection. Section 3 discusses the general requirements for protected data sharing and access control in collaborative design environments. Section 4 presents a lean information modeling and sharing mechanism for IP protection in collaborative design, where product information is shared selectively and interoperably with fine-grained access control. Section 5 shows the application of the mechanism in different data sharing scenarios and Sec. 6 discusses the advantage as well as tradeoffs of this approach.

2 Background

2.1 Digital Rights Management. Digital rights management (DRM) was created as a means of managing digital data containing intellectual property and it refers to the technologies and mechanisms specifically developed to manage digital rights. DRM includes several technology domains including security and trust, payment systems, and e-commerce system. In the entertainment industry, IP protection methodologies are driven by the needs of digital multimedia market. Some companies that provide proprietary digital rights management system (DRMS) technology include Adobe DRM [3], IBM EMMS [4], LockStream [5], and Microsoft RMS [6]. DRMS often includes centralized control that will monitor, regulate, and price each subsequent use of a computer file that contains media content, such as video, audio, photos, or text.

In DRMS, core protection technologies include encryption, passwords, watermarking, digital signature, digital fingerprint, copy detection systems, and payment systems [7]. Broadcast encryption [8] allows an encrypted message to be broadcasted so that only a dynamically changing designated group of users can decrypt it. It also enables efficient rights revocation. Watermarking [9] could offer copyright protection, ownership assertion, and integrity checks for digital content. Key issues include security, robustness, and ownership dispute. Digital contract [10] can be used in DRM for legal agreement before distribution and other related business transactions. More recently, researchers have called for DRM standards and standardization of a rights management language [11].

2.2 Watermarking and Fingerprinting. Digital watermarking is a commonly used technology in the battle against piracy. It consists in embedding in an object marks that can be used for evidencing IP ownership, channel tracing, authentication, or labeling. Watermarks are commonly used in image, audio, and video, particularly in the entertainment industry. Traditional studies on watermarking concentrate on digital multimedia content data types [12,13]. In 3D geometry data, most of the existing research focuses on data embedding in polygonal mesh models through geometry perturbation or topology change [14–16], coefficient perturbation in frequency domain [17–19], and appearance attribute change [20,21]. Similar to image and video watermarking, these methods concentrate on the appearance of 3D models and cannot be directly applied in CAD models, where modification is not tolerable. On the other hand, Ohbuchi et al. [22] developed a data embedding method for non-uniform rational B-spline (NURBS) curves and surfaces using reparametrization without changing their original geometric shape.

Fingerprinting is an alternative to watermarking that does not modify objects. It captures a design's distinctive characteristics and registers them with a trusted arbiter, who can later judge IP infringements. An effective fingerprinting technique must achieve minimal overhead, require low effort, and be secure against multi-party collusion. In electronic design, as current CAD tool and large-scale integration capabilities such as field-programmable gate arrays (FPGA) create a new market of reusable digital designs, the fingerprinting process [23,24] can produce secure marks and unique physical layouts for each design instance recipient. In shape design, Ko et al. [25] developed matching methods between two NURBS surfaces based on integral properties and Gaussian and mean curvatures.

2.3 Access Control. Access control is one of the most effective mechanisms used to enhance data security and IP protection. Major access control methods include: (1) *discretionary access control* (DAC) [26], where access is based solely on the identity of the person trying to access the data resource; (2) *mandatory access control* (MAC) [27], where mechanisms assign security levels to data resources and security clearances to users, and users have access only to data for which they have clearance; and (3) *role-based access control* (RBAC) [28], where permissions are granted based on *roles*. A role is a job function with associated authority and responsibility that are specified within the context of an organization.

No single access control mechanism can provide the greatest overall benefit to all users in all circumstances. Tradeoffs have to be made regarding performance, compatibility, and ease of use for the quality of protection in a particular situation. RBAC models are prescribed as a generalized approach for access control and policy neutral [29]. Role hierarchies and constraints enable a wide range of security policies to be expressed. Some extensions of the RBAC model also exist [30,31].

In addition, cryptography is valuable in access control of data independent of system implementations. *Cryptographic access control* (CAC) is the mechanism to control data access by key distribution, which is flexible to operate across multiple administrative domains and heterogeneous security policies. Harrington and Jensen [32] proposed a CAC model to ensure secrecy and authenticity of distributed file systems based on asymmetric cryptography and to ensure integrity and availability by using a log-structured file system.

The role-based data access control is a central issue for data security management in distributed product development environments. Cera et al. [33,34] applied the mechanisms of RBAC to information protection in design collaboration. A central role-based view control system is developed to manage collaborators' viewing and modeling privileges. Multi-resolution geometry is generated by the methods of mesh simplification and progressive mesh. We extended RBAC to a Scheduled Role-Based Distributed Data Access Control (S-RBDDAC) model [35,36] in order to support data security management in a distributed design environment. It allows for fine-grained data access control at both the server and client sides, securing different levels of design information in collaborative design.

3 Access Control Requirements for Data Sharing in Collaborative DESIGN

Access control is an essential part of any collaborative environment to protect data from unauthorized users. It is important to determine what the data access control requirements are and relate these requirements to the information infrastructure of a collaborative product development environment. Users or entities in the collaborative process must be identified by properly established access control mechanisms before access is granted or authorization is issued. The primary objective of access control in collaborative environments is the preservation and protection of confidential information, the integrity of data, and continued availability of information, systems, and resources. Authentication validates a subject's identity, while authorization determines what resources the subject is allowed to access. Authorization also assigns privileges such as the abilities to read, write, or modify.

In addition to the above basic requirements for access control, collaborative environments possess special characteristics:

(1) *The dynamic nature of users and groups:* The activity-level security infrastructure for data exchange should be separated from the organization-level security infrastructure, since participants of inter-organizational data exchange and workflow may change dy-

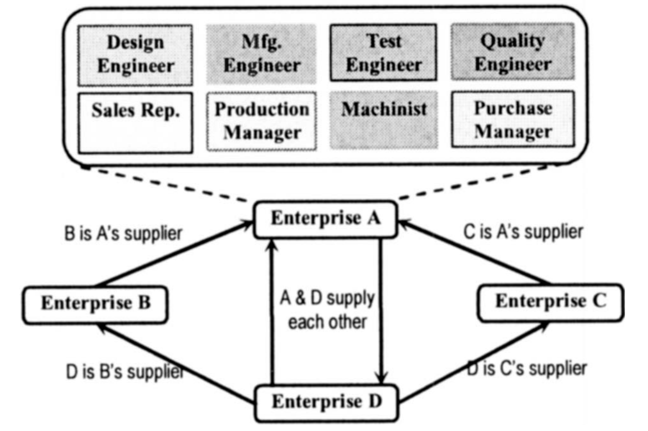


Fig. 1 An example of peer-to-peer collaboration relations among enterprises and within an enterprise

namically during the life cycle of an activity due to workflow scheduling and balancing, system and software upgrading, and corporation split, merge, and reconfiguration. Stakeholders may change during the life-cycle of a project as functional, technical, and financial needs change. New outsourcing may be needed as a project proceeds. Therefore, a flexible access control model is needed to enable distributed design data access and sharing in this environment. When an information supply chain relation is established or terminated, the access control system should easily accommodate the frequent changes.

(2) *Peer-to-peer collaboration:* Traditional access control decisions are made based on subjects-permission-object relations at the level of data files in a centralized produce lifecycle management (PLM) system. The conventional mechanism is too coarse in the situation of inter-organizational data exchange and information flow management, which tends to be multi-level and context-dependent. An original equipment manufacturer (OEM) normally has multi-tier suppliers, and a supplier participates in different design projects for multiple OEMs. Storing all design data in OEM's central PLM system is hardly acceptable to suppliers. Instead of static client-server relations, peer-to-peer relations exist in a design collaboration environment, where stakeholders share information mutually within an enterprise as well as enterprise-to-enterprise, as shown in Fig. 1. In this peer-to-peer environment, information flow is bidirectional and dynamic. Decentralized and scalable access control mechanisms that support many-to-many relations with multiple granularity requirements are necessary. This enhances data security of existing business-to-business virtual design environment such as FIPER [37,38].

(3) *Long-term and short-term trust relationship:* To maintain and manage trusted business relations in enterprise-to-enterprise collaboration, reliable access control mechanisms for data that reside at client side should be established. Only necessary information is transmitted and exposed to receivers. Different views of data and data flow based on users' need-to-know criteria and their affiliated organization should be provided. It is important to have access controls on the data that have been sent to collaborators during the process of specification exchange. The amount of information to expose to the collaborators will depend on the trust levels between collaboration participants. Within one data flow relation, there could be different projects involved, and segments of data within one single data file need to be differentiated and managed.

(4) *Large-volume data with heterogeneous formats in long and repetitive transactions:* Different from single-cycle e-business transactions, data transactions in design collaboration usually have multiple cycles involved such as in design change, collision detection, and simulation. This requires a large amount of information such as geometry and mesh data to be sent over networks repetitively. Design data may be in different formats (e.g., text description, formula, 2D or 3D CAD data, and image). Efficient and interoperable data modeling is needed to share heterogeneous design data during design collaboration.

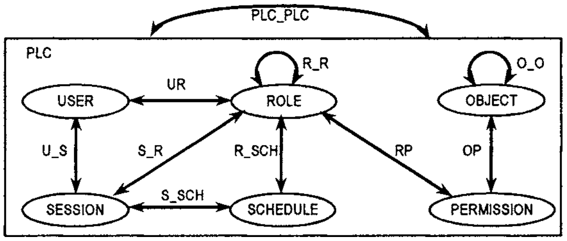


Fig. 2 S-RBDDAC model

New access control in dynamic collaborative design environments should go beyond just labeling subjects and objects. An access control model should be based on less volatile relations. Another level of abstraction such as roles, which enables indirect references, can provide more stable controls. It should also allow fine-grained information dissemination at different levels of data set (e.g., assembly, component, feature, surface, and curve) and different views of data based on how organizations and individuals behave in a task and to support the least privilege security principle, as well as the communication bandwidth limit and the frequency of data transfers and updates. Furthermore, data set level access control should consider the heterogeneity of data formats. A general model should not make too many assumptions about data formats and storage media. Data could be stored as documents or in databases. Thus, associated data models for information exchange need to provide an interoperable solution for different data types and formats.

In this paper, a secure data modeling and sharing framework for intellectual property protection is presented, which is based on the S-RBDDAC model and generic XML data format for fine-grained information sharing. Utilizing a role-oriented data access procedure within a dynamic collaboration environment reduces the complexity of security administration. To improve the flexibility further, another dimension of time/schedule is introduced such that permissions are based on what function a user performs in a particular project within a specified time period. Functional roles such as designers, manufacturers, and sales personnel within an enterprise, as well as suppliers, government agencies, retailers, and customers outside of the enterprise become the intermediate layer between users and permissions.

4 Distributed Design Data Access Control

4.1 S-RBDDAC Model. S-RBDDAC is an access control model for collaborative design data, not considering operating system level operations and processes. It intends to separate data level controls from operating system or network level controls. This model defines elements and their relations, as shown in Fig. 2 and listed in Table 1.

Within the model, the following relations are defined:

- $UR \subseteq U \times R$: a many-to-many mapping user-to-role assignment relation.
- $U\_S: U \rightarrow 2^S$, a mapping of a user to the sessions created by the user.
- $S\_R: S \rightarrow 2^R$, a mapping of a session to the involved roles.
- $RP \subseteq R \times P$: a many-to-many mapping permission-to-role assignment relation.
- $OP \subseteq O \times P$: a many-to-many mapping object-to-permission assignment relation.
- $S\_\mathrm{SCH}: S \rightarrow \mathrm{SCH}$, a one-to-one mapping of a session to its schedule.

Table 1 Elements of S-RBDDAC

Elements	Definitions
Policy (PLC)	Operating rules that can be referred to as a means of maintaining order, control, and consistency for ease of management.
User (U)	A human being; however, the concept of a user can be extended to machines, networks, or intelligent autonomous agents.
Session (S)	An activity or work process.
Role (R)	A job function with associated authority and responsibility that are specified within the context of an organization.
Object (O)	Any data resource or data segment subject to access control.
Permission (P)	An approval to access to one or more protected objects.
Schedule (SCH)	A collection of access time intervals, locations, and collaboration states.

Table 2 Privileges and states

Terms	Definitions
Data set (d)	A set of data containing fine-grained information.
Instance (I)	A snapshot of a session.
Value-adding collaboration (v)	Collaboration in which value is added during the session.
non-value-adding collaboration (nv)	Collaboration in which value is not added during the session.
UP State (UP)	The state in which the user is allowed access for expected value-adding collaboration.
DOWN State (DOWN)	The state in which the user is not allowed access for expected non-value-adding collaboration.
Negative State (NEG)	The state in which the user is strictly denied access during an entire session.
Positive privilege	The positive privilege contains the UP and DOWN states; this privilege exists for all collaborators who have some type of access to a particular resource.
Negative privilege	The negative privilege contains the Negative state; this privilege exists for all collaborators who do not have any type of access to a particular resource.

- $R\_\mathrm{SCH}: R \rightarrow 2^{\mathrm{SCH}}$, a mapping of a role to its action schedules.
- $RR \subseteq R \times R$: a partially ordered role hierarchy.
- $OO \subseteq O \times O$: a partially ordered object hierarchy.
- $\mathrm{PLC\_PLC} \subseteq \mathrm{PLC} \times \mathrm{PLC}$: a partially ordered policy hierarchy.

The user-to-role assignment allows the same user to play different roles, and a single role to be assigned to a team of users. The permission-to-role assignment allows a single permission to be applicable for multiple roles. One role can have multiple tasks. The object-to-permission assignment ensures that an object can be accessed with different levels of granularity.

The role hierarchy is mathematically a partial order defining a *seniority* relation between roles, whereby senior roles acquire the permissions of their juniors, and junior roles acquire the user membership of their seniors. The object hierarchy defines a *subset* relation between objects, whereby accessing subset objects at least needs permissions of accessing their supersets. The policy hierarchy is a policy *delegation* relation, in which a delegated policy should be stricter than its parent policies. The hierarchical relations of roles, objects, and policies are illustrated in Fig. 3.

This model eliminates unnecessary access from both functional and time considerations. It provides two-dimensional access controls such that access is granted only *when* absolutely necessary and collaboration is established only *where* needed.

User access of the data needs to be constrained by time. For example, a manufacturer within a specified team is not allowed to access relevant data until all priori time schedules have been met. This scheduled access control procedure provides a *lean* approach to data set (d) access. Table 2 provides definitions of some more terms in the S-RBDDAC model.

In any given instance (I) of collaboration, only value-adding collaboration (v) is needed during the process of a product development project. Non-value-adding collaboration (nv), such as unnecessary data browsing, redundant service request, and unscheduled service initiation, interferes with other collaboration. Figure 4 depicts the relationships and transitions between the states as well as privileges. The project's owner creates sessions and configures users, roles, schedules, permissions, and objects within the sessions. The *schedule* element contains a set of time intervals and locations with associated collaboration status. According to the schedule, roles (and corresponding data access privileges) are classified as being in either the UP state (UP: the access is permitted) or DOWN state (DOWN: the access is not allowed) as the case may be for the corresponding session. A *negative* state is the situation where a user's access to a particular resource is denied for the entire session.

A role access privilege is UP when the user's collaborative expertise is required for the current stage. Such collaboration may be required at different stages of the collaborative effort within a product design and development project session. The ordinary stateless role-based model without schedule may cause access clutter. The S-RBDDAC model reduces the granted positive privi-

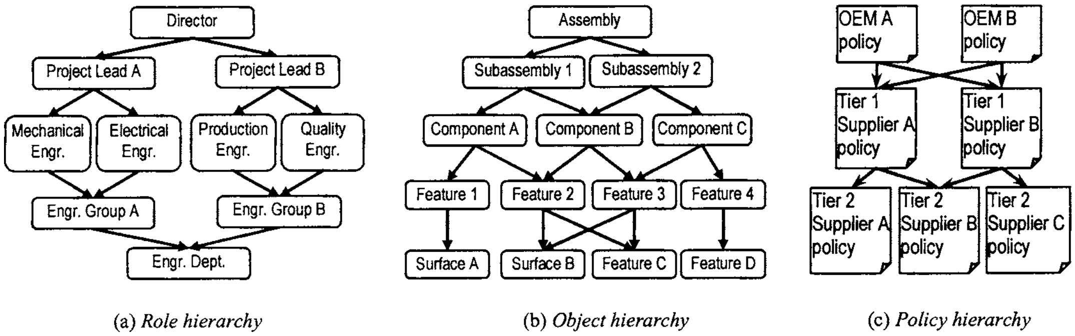


Fig. 3 Hierarchical relations of roles, objects, and policies

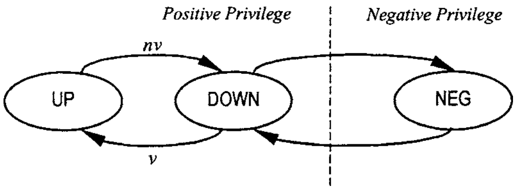


Fig. 4 Transitions between states and privileges

leges to the lowest level. Depending on the requirements of the project, accesses are scheduled as needed. Constraints can be added to the S-RBDDAC model, which are *separation of duty* (SOD) *relations* to enforce conflict of interest policies. These policies may be used to prevent users from exceeding a reasonable level of authority. The constraints can be added to any relations such as user-to-role assignments, role hierarchies, permission-to-role assignments, policy delegation, and SOD relations. For example, users from two competing suppliers cannot be assigned roles that work in one common session; a junior role cannot inherit two senior roles that have mutually exclusive permissions; a project owner may delegate its policy associated with the project to subordinate collaborative partners in order to ensure compliance, yet maintaining some level of security control of the project; and a junior role inherits SOD constraints from its senior roles. SOD relations can be dynamic, and do not persist as schedules change. Figure 5 shows an example of access control policies for two collaborating corporations, each of which creates its own policy to protect its design data based on the defined roles, scheduled time intervals, as well as locations for each session.

4.2 Lean Information Sharing through XML. The structures and sizes of data involved in the whole product development cycle could vary significantly. Different data types need to be shared, including specification, CAD geometry, mesh model, simulation code and result, image, as well as documents containing text, graphs, formulas, etc. This puts a formidable challenge on data access control and management. Organizations find it difficult comparing and sharing data with other enterprise data sources due to varying data formats as depicted in Fig. 6(*a*). A ubiquitous format standard will simplify the format transformation process. Figure 6(*b*) shows the integration of the different design data sources into one common format. Therefore, to enable access control for different product data in various formats, an indirect approach has to be taken in a distributed data-sharing environment. Instead of directly sending original data, a common protocol such as an XML interface for different data formats may be established in advance. XML provides a common syntax for modeling data. It offers a user-defined and extensible format to represent data and information for different application areas. XML also allows for separation of content from format, enabling the processing and presentation of information. As an emerging data exchange standard, XML can handle complex data structures (such as vector graphics, e-commerce transactions, mathematical equations) and provides interoperability, system-independence, ease of transformation and data parsing.

While proprietary data format is the major hurdle for engineering data exchange, open standards are needed to allow interoperable data sharing. However, the special syntax and schema definitions of current STEP/EXPRESS standards build high development and implementation barrier in industry practice [39]. In contrast, XML is easily extensible and numerous software tools are available with little cost. In order to take advantage of XML's flexibility and popularity in development community, ISO is adopting XML into STEP standards as part 28 [40].

XML is an open medium for information transferring and sharing. Securing data in this format is critical to protect IP. Some related XML security standards are being developed. For example, the XML Access Control Markup Language (XACML) [41] communicates policy information regarding data set access control, specifying what portions of the data can be exposed to appropriate parties. XML Signature, XML Encryption, and XML Key Management [42] schemes address varying requirements for access authority, confidentiality and data integrity.

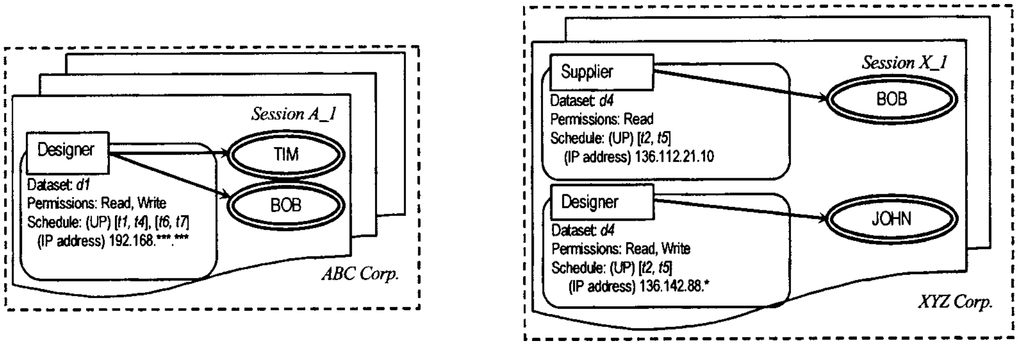
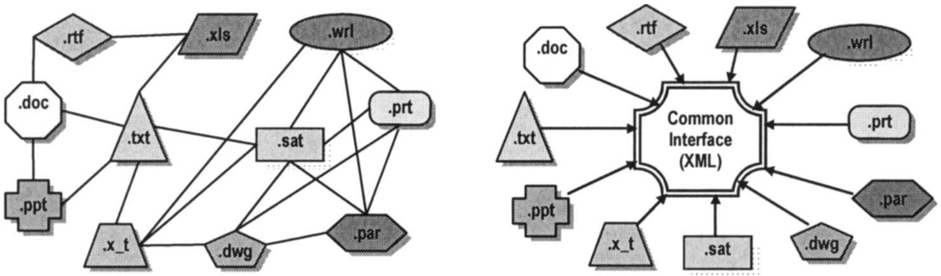


Fig. 5 Two examples of access control policies



(a) up to $O(n^2)$ conversions are needed for *n* different data formats

(b) common interface reduces the number of conversions / translations

Fig. 6 Different data sources/formats interaction with a common data interface

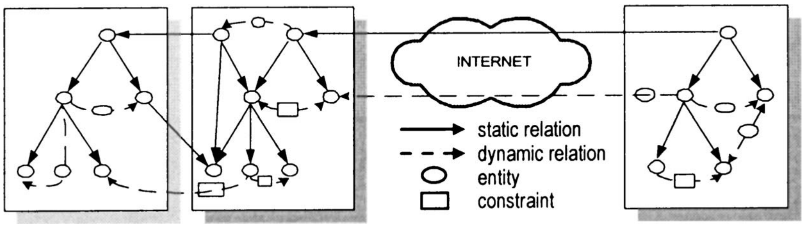
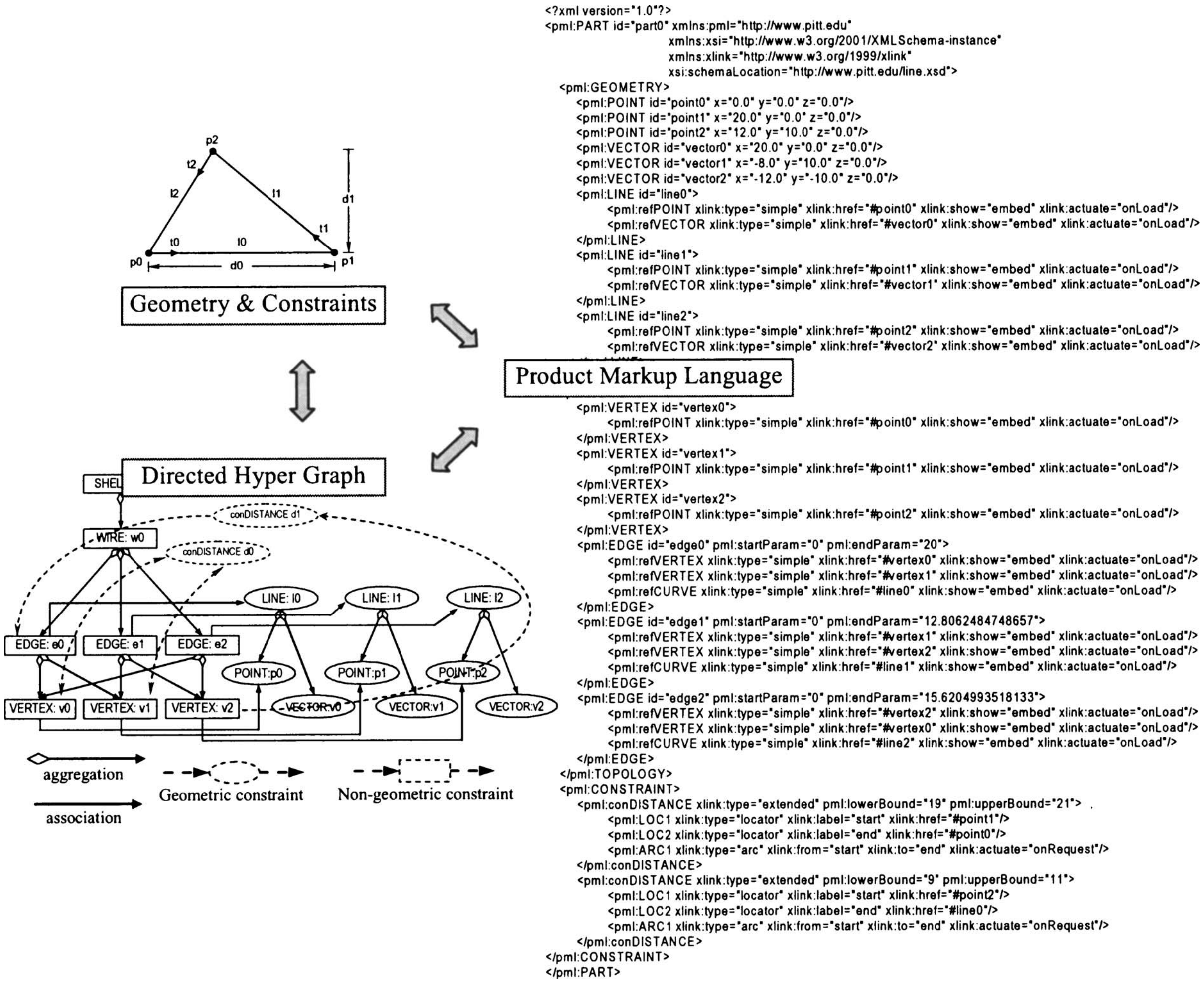


Fig. 7 Universal linkage between files with static (aggregation, generalization, association) and dynamic (geometric and non-geometric constraint) relations across file boundaries

We developed a distributed design data scheme, Universal Linkage-Product Markup Language (UL-PML), to model geometry, topology, features, and constraints in networked collaborative design environments. Unlike current open standard STEP and other industry standards for visualization (e.g., JT-Open [43], X3D [44]) and data exchange (e.g., PLM-XML [45]), UL-PML is a constraint-enabled distributed product data model in native XML format. Multidisciplinary design information can be captured, distributed, and linked with different levels of granularity and flexibility. UL-PML scheme captures geometric and non-geometric relations among entities in a virtual link style so that references between entities can be made across the boundary of files and physical locations in a distributed design environment. This scheme allows design information to be integrated in a collaborative design environment. Besides static relations among design objects, dynamic relations/constraints are also incorporated. Detailed description of UL-PML and the comparison with other representations are discussed in Ref. [46].

The network-aware data model intends to improve design *information* interoperability based on general *data* interoperability. XML syntax guarantees the openness of the UL model. The standardization of PML schemas can additionally provide semantics-level interoperability.

The UL model does not require that one data file contain all the information relevant to the designed product. Incorporating physical distribution and logical integration, it makes partial design information storage and retrieval easy to realize. This provides another level of granularity and increases the flexibility during design information query. Design information can be stored modularly without compromising the integrity of the whole prod-



```
<?xml version="1.0"?>
<pml:PART id="part0" xmlns:pml="http://www.pitt.edu"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xmlns:xlink="http://www.w3.org/1999/xlink"
                    xsi:schemaLocation="http://www.pitt.edu/line.xsd">
  <pml:GEOMETRY>
    <pml:POINT id="point0" x="0.0" y="0.0" z="0.0"/>
    <pml:POINT id="point1" x="20.0" y="0.0" z="0.0"/>
    <pml:POINT id="point2" x="12.0" y="10.0" z="0.0"/>
    <pml:VECTOR id="vector0" x="20.0" y="0.0" z="0.0"/>
    <pml:VECTOR id="vector1" x="-8.0" y="10.0" z="0.0"/>
    <pml:VECTOR id="vector2" x="-12.0" y="-10.0" z="0.0"/>
    <pml:LINE id="line0">
        <pml:refPOINT xlink:type="simple" xlink:href="#point0" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refVECTOR xlink:type="simple" xlink:href="#vector0" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:LINE>
    <pml:LINE id="line1">
        <pml:refPOINT xlink:type="simple" xlink:href="#point1" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refVECTOR xlink:type="simple" xlink:href="#vector1" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:LINE>
    <pml:LINE id="line2">
        <pml:refPOINT xlink:type="simple" xlink:href="#point2" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refVECTOR xlink:type="simple" xlink:href="#vector2" xlink:show="embed" xlink:actuate="onLoad"/>
    <pml:VERTEX id="vertex0">
        <pml:refPOINT xlink:type="simple" xlink:href="#point0" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:VERTEX>
    <pml:VERTEX id="vertex1">
        <pml:refPOINT xlink:type="simple" xlink:href="#point1" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:VERTEX>
    <pml:VERTEX id="vertex2">
        <pml:refPOINT xlink:type="simple" xlink:href="#point2" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:VERTEX>
    <pml:EDGE id="edge0" pml:startParam="0" pml:endParam="20">
        <pml:refVERTEX xlink:type="simple" xlink:href="#vertex0" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refVERTEX xlink:type="simple" xlink:href="#vertex1" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refCURVE xlink:type="simple" xlink:href="#line0" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:EDGE>
    <pml:EDGE id="edge1" pml:startParam="0" pml:endParam="12.8062484748657">
        <pml:refVERTEX xlink:type="simple" xlink:href="#vertex1" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refVERTEX xlink:type="simple" xlink:href="#vertex2" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refCURVE xlink:type="simple" xlink:href="#line1" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:EDGE>
    <pml:EDGE id="edge2" pml:startParam="0" pml:endParam="15.6204993518133">
        <pml:refVERTEX xlink:type="simple" xlink:href="#vertex2" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refVERTEX xlink:type="simple" xlink:href="#vertex0" xlink:show="embed" xlink:actuate="onLoad"/>
        <pml:refCURVE xlink:type="simple" xlink:href="#line2" xlink:show="embed" xlink:actuate="onLoad"/>
    </pml:EDGE>
  </pml:TOPOLOGY>
  <pml:CONSTRAINT>
    <pml:conDISTANCE xlink:type="extended" pml:lowerBound="19" pml:upperBound="21"> .
        <pml:LOC1 xlink:type="locator" xlink:label="start" xlink:href="#point1"/>
        <pml:LOC2 xlink:type="locator" xlink:label="end" xlink:href="#point0"/>
        <pml:ARC1 xlink:type="arc" xlink:from="start" xlink:to="end" xlink:actuate="onRequest"/>
    </pml:conDISTANCE>
    <pml:conDISTANCE xlink:type="extended" pml:lowerBound="9" pml:upperBound="11">
        <pml:LOC1 xlink:type="locator" xlink:label="start" xlink:href="#point2"/>
        <pml:LOC2 xlink:type="locator" xlink:label="end" xlink:href="#line0"/>
        <pml:ARC1 xlink:type="arc" xlink:from="start" xlink:to="end" xlink:actuate="onRequest"/>
    </pml:conDISTANCE>
</pml:CONSTRAINT>
</pml:PART>
```

Fig. 8 UL model representation and mapping

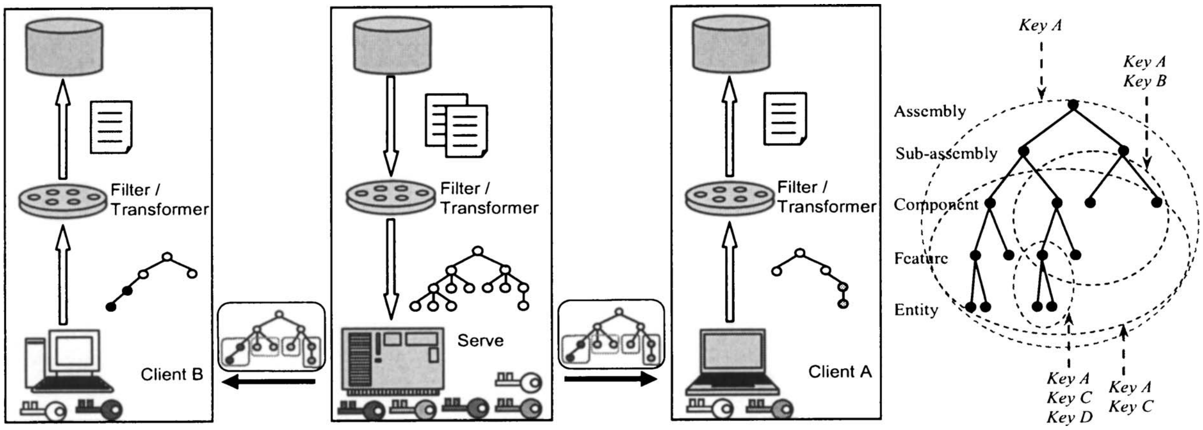


Fig. 9 Selective information flow based on XML encryption with different key sets corresponding to various subsets of data

uct. This reduces the requirement for computational time and storage space. Hence, it increases flexibility for scalable designer systems, and encourages reuse of design at subassembly, component, feature, or surface levels.

Lean information sharing and exchange for collaborative design can be realized over the Internet. Relations of design data elements and constraints are represented in the UL model to create an information framework. They are explicit linkages that ensure product data's consistency, thus a logically integrated set of design information can be built in a distributed environment. The relations among entities are not restricted within one data file. As illustrated in Fig. 7, relations of entities (both static and dynamic) in different domains and physical locations can be created. One can easily refer entities in other data files, either at the same machine or other locations over the Internet. This allows partial data to be transferred over networks without compromising logical integrity.

The integrated geometric and non-geometric constraint representation in UL model incorporates more design knowledge into design data. The explicit capturing of multidisciplinary constraints, especially non-geometric constraints, enables a more complete information representation than current standard formats. This provides a more comprehensive support for design intent representation at different design stages. Graphically, the UL model can be represented by Directed Hyper Graphs. Textually, the UL model is stored as PML. Figure 8 shows the model representation of the UL-PML scheme.

The typical design data have a hierarchical structure. This naturally fits into the XML tree structure. Detailed geometry and topology in a design can also be mapped to PML tree, which strictly follows the syntax of XML. The compliance to industrial computation and communication standard is the premise of computational interoperability at the syntax level.

Nevertheless, there are open issues in applying XML to product data representation. First, the mapping between existing CAD data standards and the XML structure needs to be standardized [47] (ISO TC184/SC4 committee is working on this issue). The XML schema for product data needs to be properly defined according to current needs as well as future extensions. Second, the XML syntax is not as succinct as other CAD data formats. The size of XML file is relatively large, and redundancy exists in tagged text. Third, the flexibility of XML syntax makes standardization difficult. Issues include child elements versus attributes, early-binding versus late-binding [48].

4.3 Fine-Grained Control for Shared Data Set. In a distributed environment, cryptography is ideal for data set level access control. A data set could be sent to multiple collaborators who have different privileges to access the data subsets. It is also possible that a subset of the data set that the collaborator received would be sent to a third party with supply chain relationships. The access privilege is granted to each role through key distribution and policy delegation. Policies determine who uses the keys and when these keys come into effect. The number of keys a collaborator owns is corresponding to the permission he or she has been granted. The policy applied for first-tier collaborators can be delegated to other tier collaborators.

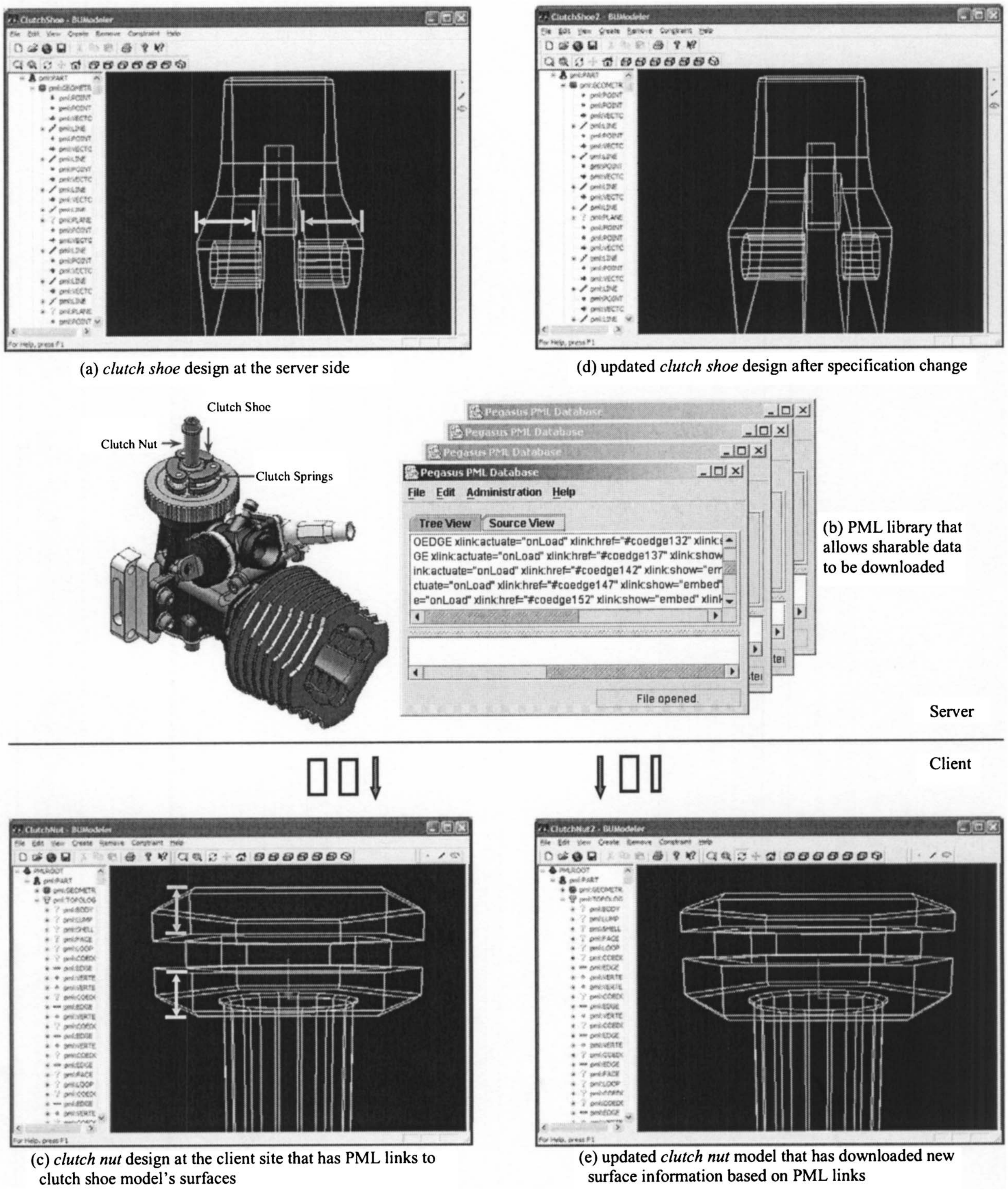
The key should be established before communication can begin. The key distribution scheme should be secure against known key attacks. If a particular session key is compromised, it should not affect the usage of other session keys. As shown in Fig. 9, the XML-based cross-domain information model becomes a bridge between different data types used in various product development areas. Different key sets may be distributed to users at different security levels such that different sets of data are unlocked and different views of data are provided.

The S-RBDDAC model needs two categories of mechanisms for implementation. One is at the system administration level, and the other is at the data set level. Each project owner defines and implements the owner's access control policy based on its interests. No centralized policy enforcement exists for data access control. At the system level, accesses to memory, disks, database, and other data media need to be controlled. Privileges are defined in security policies and policies are enforced through mechanisms of locks, synchronization, and file read/write protections based on access control matrices. Although perfect security is not possible, we can achieve computational security if the cost of breaking the cipher is more than the value of the information it is protecting and the time required to break the encryption exceeds the useful lifetime of the protected information.

5 Data Sharing Scenarios

At the initiation of a design project, the owner of the project sets up administrative protocols such as role assignments and data security levels based on contracts or other prior legal discussions. Data ownership and information tracking also have to be determined and agreed upon. The S-RBDDAC model enables fine-grained access control on design data through role assignments and hierarchies of roles, data sets, policies, as well as schedules. Recursive encryption with different keys guarantees minimal data exposure. Partial data sharing through XML prevents reverse engineering. All of these mechanisms provide different dimensions of IP protection. This approach generally supports both geometric and non-geometric design data sharing. Such interoperable integration is possible as a result of the XML common interface.

5.1 Selective Data Exchange. In selective data exchange,



```
OEDGE xlink:actuate="onLoad" xlink:href="#coedge132" xlink:s
GE xlink:actuate="onLoad" xlink:href="#coedge137" xlink:show
ink:actuate="onLoad" xlink:href="#coedge142" xlink:show="em
ctuate="onLoad" xlink:href="#coedge147" xlink:show="embed"
e="onLoad" xlink:href="#coedge152" xlink:show="embed" xlink
```

(a) *clutch shoe* design at the server side

(d) updated *clutch shoe* design after specification change

(b) PML library that allows sharable data to be downloaded

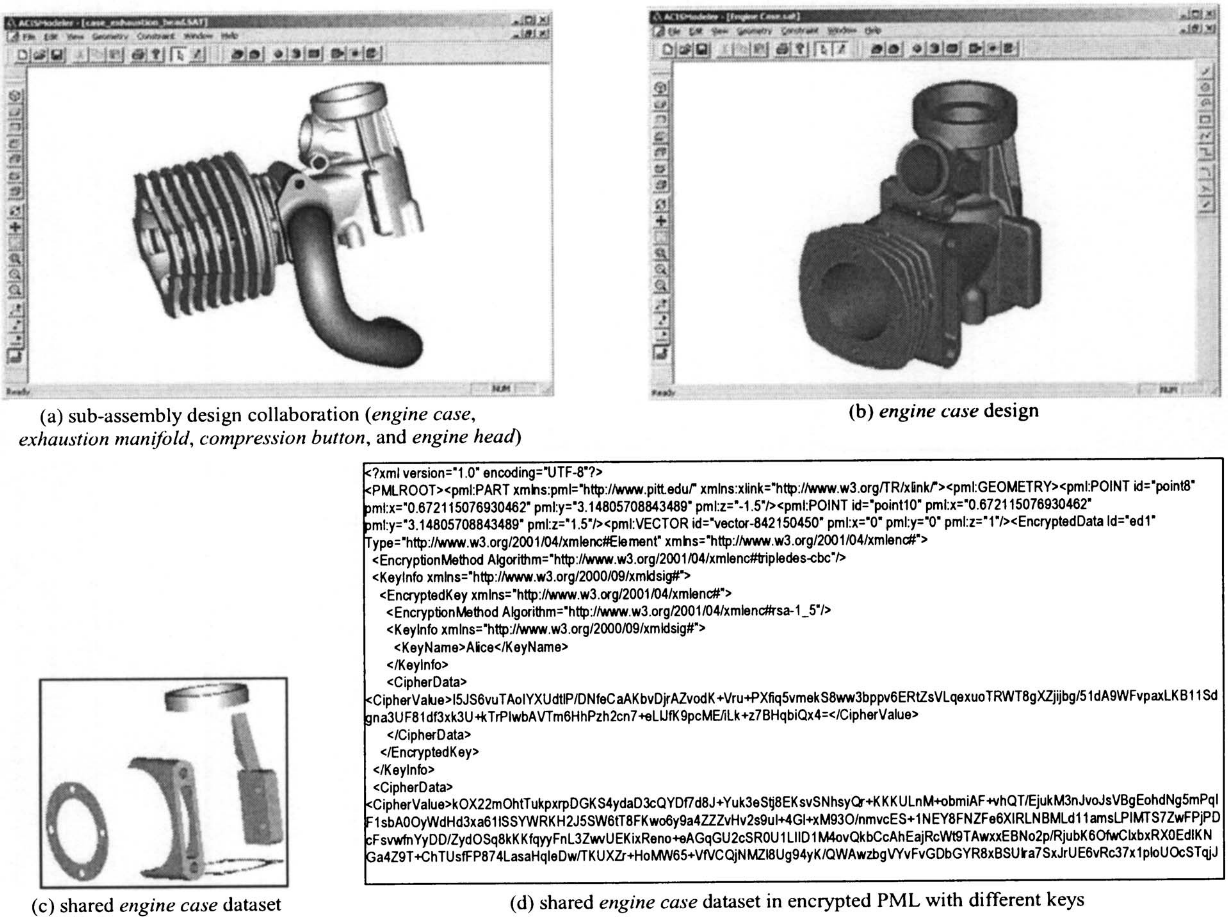
(c) *clutch nut* design at the client site that has PML links to clutch shoe model's surfaces

(e) updated *clutch nut* model that has downloaded new surface information based on PML links

Fig. 10 Lean information exchange of engine design models in UL-PML scheme

only necessary and relevant geometry, parameters, and constraints are transmitted based on the PML model's fairly loose structure. In an engine design example shown in Fig. 10, two groups who design clutch shoes and clutch nuts need to share some data to make sure that the contacting surfaces of the two parts geometrically match each other. PML links between surfaces in two components can be built. The geometry and topology information of the contacting faces in one can be fetched from the other to maintain consistency. In this linkage example, the clutch shoe (Fig. 10(*a*)) is at the server site. Once the sharable data in PML format is published in a library (Fig. 10(*b*)), it is available for the clutch nut at the client site as reference (Fig. 10(*c*)). Instead of transferring the whole data file, only relevant surfaces are transferred through data sharing agents based on Common Object Request Broker Architecture (CORBA). If any change is made at the server site (Fig. 10(*d*)), the client can download the new geometry through relevant PML links to maintain consistency (Fig. 10(*e*)).

5.2 Geometric Data Sharing with Multiple Views. XML encryption provides end-to-end security for structured data trans-



(a) sub-assembly design collaboration (*engine case*, *exhaustion manifold*, *compression button*, and *engine head*)

(b) *engine case* design

```
<?xml version="1.0" encoding="UTF-8"?>
<PMLROOT><pml:PART xmlns:pml="http://www.pitt.edu/" xmlns:xlink="http://www.w3.org/TR/xlink/"><pml:GEOMETRY><pml:POINT id="point8"
pml:x="0.672115076930462" pml:y="3.14805708843489" pml:z="-1.5"/><pml:POINT id="point10" pml:x="0.672115076930462"
pml:y="3.14805708843489" pml:z="1.5"/><pml:VECTOR id="vector-842150450" pml:x="0" pml:y="0" pml:z="1"/><EncryptedData Id="ed1"
Type="http://www.w3.org/2001/04/xmlenc#Element" xmlns="http://www.w3.org/2001/04/xmlenc#">
 <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
 <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
  <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
   <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
   <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <KeyName>Alice</KeyName>
   </KeyInfo>
   <CipherData>
<CipherValue>I5JS6vuTAoIYXUdtlP/DNfeCaAKbvDjrAZvodK+Vru+PXfiq5vmekS8ww3bppv6ERtZsVLqexuoTRWT8gXZjijbg/51dA9WFvpaxLKB11Sd
gna3UF81df3xk3U+kTrPIwbAVTm6HhPzh2cn7+eLIJfK9pcME/iLk+z7BHqbiQx4=</CipherValue>
   </CipherData>
  </EncryptedKey>
 </KeyInfo>
 <CipherData>
<CipherValue>kOX22mOhtTukpxrpDGKS4ydaD3cQYDf7d8J+Yuk3eStj8EKsvSNhsyQr+KKKULnM+obmiAF+vhQT/EjukM3nJvoJsVBgEohdNg5mPql
F1sbA0OyWdHd3xa61ISSYWRKH2J5SW6tT8FKwo6y9a4ZZZvHv2s9ul+4GI+xM93O/nmvcES+1NEY8FNZFe6XIRLNBMLd11amsLPIMTS7ZwFPjPD
cFsvwfnYyDD/ZydOSq8kKKfqyyFnL3ZwvUEKixReno+eAGqGU2cSR0U1LIID1M4ovQkbCcAhEajRcWt9TAwxxEBNo2p/RjubK6OfwClxbxRX0EdIKN
Ga4Z9T+ChTUsfFP874LasaHqleDw/TKUXZr+HoMW65+VfVCQjNMZI8Ug94yK/QWAwzbgVYvFvGDbGYR8xBSUIra7SxJrUE6vRc37x1ploUOcSTqjJ
```

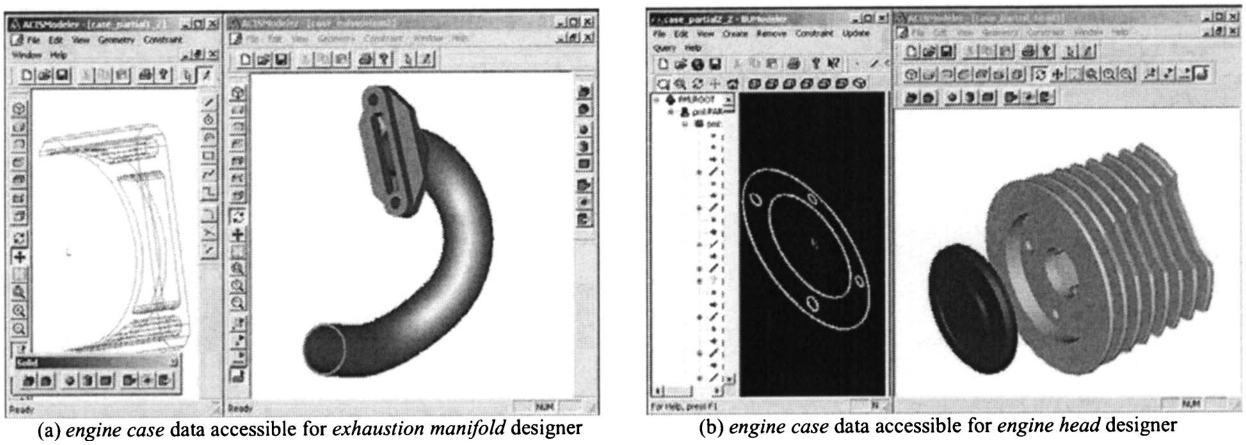
(c) shared *engine case* dataset

(d) shared *engine case* dataset in encrypted PML with different keys

Fig. 11 Selective geometry sharing in encrypted PML

fer. Based on the XML encryption scheme, relevant information is extracted from original types at the data owner's site and transformed into XML format. This XML data set then can be transmitted to different client sites securely by encryption mechanisms. Once the data arrive at a client site, it can be transformed into the original or a different format and processed locally. The application of the S-RBDDAC Model to PML restricts what portion of a PML document a client can see and when such access is permitted. Such restrictions are achieved through key management. Encrypted PML provides nodal confidentiality for product design data through the elimination of seemingly unnecessary nodes for any given instance or session. The entire PML file can be encrypted. Alternatively, a portion of the PML tree may be encrypted, such as an element of the tree or only the contents of the element.

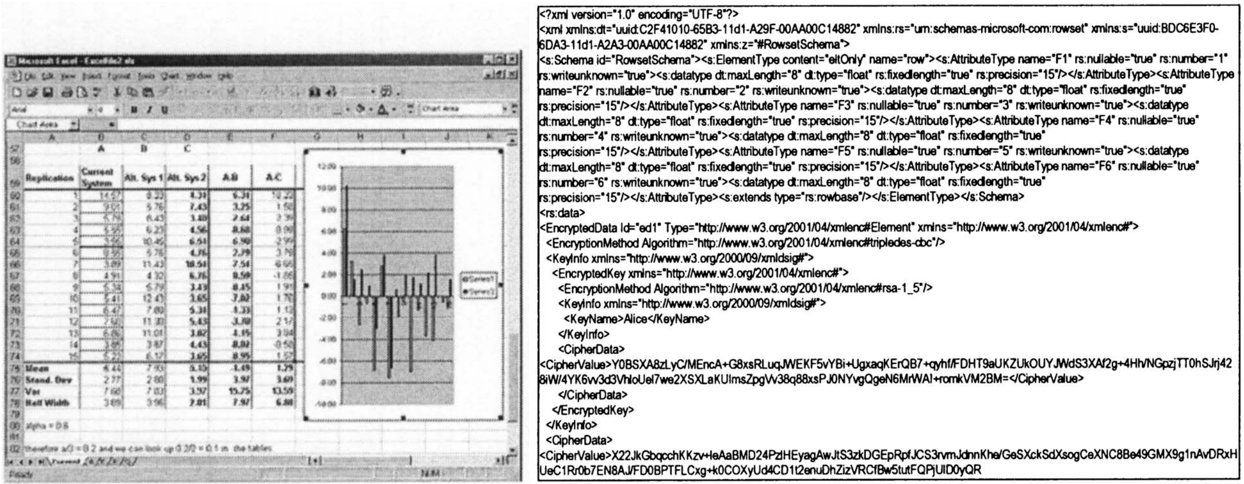
Multiple views of the same data set can be achieved by encryption recursively with different keys. As shown in Fig. 11, the design of an engine sub-assembly is distributed among three companies, A, B, and C. While the engine case is designed at A, its geometric interfaces with other components need to be shared with B and C to ensure proper assembly relation. As the owner of the engine case model, A defines security policy to share engine



(a) *engine case* data accessible for *exhaustion manifold* designer

(b) *engine case* data accessible for *engine head* designer

Fig. 12 Different views of shared data provided for different roles



Microsoft Excel - Excelfile2.xls

	A	B	C		
Replication	Current System	Alt. Sys 1	Alt. Sys 2	A-B	A-C
1	14.57	8.23	4.31	6.34	10.22
2	9.01	5.76	7.43	3.25	1.58
3	[illegible]	8.43	1.89	2.64	[illegible]
4	5.55	6.23	4.56	-0.68	0.99
5	[illegible]	10.46	6.54	-6.90	-2.99
6	[illegible]	5.76	4.76	2.79	3.79
7	3.89	11.43	10.54	-7.54	-6.65
8	4.91	4.32	6.76	0.59	-1.86
9	5.34	5.79	3.43	-0.45	1.91
10	5.41	12.43	3.65	-7.02	1.76
11	6.47	7.80	5.34	-1.33	1.13
12	[illegible]	11.30	5.43	-3.70	2.17
13	6.86	11.01	3.82	-4.15	3.04
14	3.85	3.87	4.43	-0.02	-0.58
15	[illegible]	6.17	3.65	-0.95	1.57
Mean	6.44	7.93	5.15	-1.49	1.29
Stand. Dev	2.77	2.80	1.99	3.97	3.69
Var	7.68	7.83	3.97	15.75	13.59
Half Width	3.89	3.96	2.81	7.97	6.88

alpha = 0.6

therefore a/3 = 0.2 and we can look up 0.2/2 = 0.1 in the tables

(a) *Experimental data in Excel file*

```
<?xml version="1.0" encoding="UTF-8"?>
<xml xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882" xmlns:rs="urn:schemas-microsoft-com:rowset" xmlns:s="uuid:BDC6E3F0-
6DA3-11d1-A2A3-00AA00C14882" xmlns:z="#RowsetSchema">
<s:Schema id="RowsetSchema"><s:ElementType content="eltOnly" name="row"><s:AttributeType name="F1" rs:nullable="true" rs:number="1"
rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true" rs:precision="15"/></s:AttributeType><s:AttributeType
name="F2" rs:nullable="true" rs:number="2" rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:AttributeType name="F3" rs:nullable="true" rs:number="3" rs:writeunknown="true"><s:datatype
dt:maxLength="8" dt:type="float" rs:fixedlength="true" rs:precision="15"/></s:AttributeType><s:AttributeType name="F4" rs:nullable="true"
rs:number="4" rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:AttributeType name="F5" rs:nullable="true" rs:number="5" rs:writeunknown="true"><s:datatype
dt:maxLength="8" dt:type="float" rs:fixedlength="true" rs:precision="15"/></s:AttributeType><s:AttributeType name="F6" rs:nullable="true"
rs:number="6" rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:extends type="rs:rowbase"/></s:ElementType></s:Schema>
<rs:data>
<EncryptedData Id="ed1" Type="http://www.w3.org/2001/04/xmlenc#Element" xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
  <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
   <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
     <KeyName>Alice</KeyName>
    </KeyInfo>
    <CipherData>
<CipherValue>Y0BSXA8zLyC/MEncA+G8xsRLuqJWEKF5vYBi+UgxaqKErQB7+qyhf/FDHT9aUKZUkOUYJWdS3XAf2g+4Hh/NGpzjTT0hSJrj42
8iW/4YK6vv3d3VhloUel7we2XSXLaKUlmsZpgVv38q88xsPJ0NYvgQgeN6MrWAI+romkVM2BM=</CipherValue>
    </CipherData>
   </EncryptedKey>
  </KeyInfo>
  <CipherData>
<CipherValue>X22JkGbqcchKKzv+leAaBMD24PzlHEyagAwJtS3zkDGEpRpfJCS3rvmJdnnKhe/GeSXckSdXsogCeXNC8Be49GMX9g1nAvDRxH
UeC1Rr0b7EN8AJ/FD0BPTFLCxg+k0COXyUd4CD1t2enuDhZizVRCfBw5tutFQPjUID0yQR
```

(b) *Encrypted XML for data exchange*

Fig. 13 Secure selective non-geometric engineering data exchange by XML encryption

case data. Engineers at A can select the specific shared interface data sets based on technical requirements of assembly, transform them into PML, and encrypt with different keys. According to the policy, different sets of keys are distributed to B and C who design exhaustion manifold and engine head, respectively. They have access to the minimal data sets of the interfaces related to their design, and different views of the same data can be provided for them, as shown in Fig. 12.

5.3 Non-Geometric Engineering Data Sharing. As XML format transformation becomes available for major commercial software tools, other non-geometric engineering data can be shared selectively in the same way using the XML-based lean information sharing mechanism. For example, a set of experimental data for a design (Fig. 13(*a*)) originally in Microsoft Excel spreadsheet can be easily converted into XML, and transmitted in encrypted format (Fig. 13(*b*)). Encrypted XML provides a generic and secure data structure for heterogeneous models in collaborative design. Data interoperability and openness enhance the overall information infrastructure of collaboration environment.

6 Discussion

The combination of the two mechanisms, partial data sharing and fine-grained access control, provides a multi-level security procedure for IP protection. Partial data sharing prevents reverse engineering, while scheduled access control for data set allows the protection of data segments. This lean information modeling and sharing framework supports engineering data security management at both the server and client sides in a peer-to-peer environment.

Compared to the traditional data modeling and distribution method, the lean approach may be a more effective alternative in IP protection. The use of extended S-RBDDAC model access control and distributed XML data formats can reduce the overall risks of IP infringement and improve system performance. There are costs associated with this scheme. The S-RBDDAC model increases the complexity of access control management at the server side. The overhead of scheduling and key distribution is increased in the fine-grained approach. Second, the size of XML could be four to eight times larger than original ASCII files (e.g., SAT, X_T, and STEP) that contain the same amount of information. However, the lean data exchange approach will alleviate the problem when partial data are transferred.

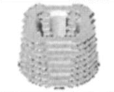
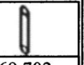
An experiment was conducted to measure and compare file sizes resulting from the traditional methodology and the lean method. Traditionally, the entire file has to be encrypted and sent to each recipient. Besides the risk of exposing unnecessary data to irrelevant parties, this also entails encrypting large files and sending across the transmission channel. In contrast, the lean approach provides customized data views for each recipient. The comparison of the data sizes is shown in Table 3.

Some other performance tradeoffs must also be considered. The transformation of the original file into XML format and segmentation of the data file into subsets are non-trivial. This will add additional overhead to the performance of the model. However, segmented subsets can be stored or cached and thus can be updated and reused with associated model privileges.

7 Conclusion

This paper presents a secure data modeling and distribution framework to protect intellectual property in collaborative design environments based on an S-RBDDAC model. This model extends the traditional RBAC model for fine-grained data access control such that lean and secure data exchange and sharing are supported. Based on the functional roles and schedules, relatively stable and easy access control for collaborative environments can be created. The uniqueness of this model includes the consideration of time, scheduling, and value-adding activity with roles, policy delegation relation in a distributed context, and fine-grained access control at data set level. Heterogeneous data are exchanged through XML common interface, which provides a neutral format to enhance data interoperability. Partial data exchange is supported in this distributed data modeling scheme to

Table 3 Size comparisons between the encrypted complete model and the partial geometry in an experiment

Complete Model			
SAT file (bytes) **Encrypted SAT (3DES/Blowfish)**	2,446,784 3,313,388	1,536,583 2,080,813	604,524 604,544
Partial Geometry			
XML file (bytes) **XML Encryption (bytes)**	83,811 112,655	76,892 103,427	69,702 93,847
Reduction of shared data size	97%	95%	84%

prevent reverse engineering. These factors increase the flexibility of data sharing and promote a secure and interoperable information infrastructure.

Acknowledgment

The authors are grateful for the support from NSF as well as Industry and Government research partners and collaborators. We also would like to thank Dr. Plamen Bliznakov of PTC, Dr. Michael Carroll of ANSYS, and Joshua Lubell of NIST for discussions, and Dr. David Lee of Northrop Grumman for encouragement, as well as the anonymous reviewers for comments to improve the paper.

References

[1] The white paper of U.S. National Science Foundation workshop on e-product design and realization for mechanically engineered products, October 19–20, 2000, Pittsburgh, PA, http://www.e-designcenter.info

[2] Parunak, H. V. D., 1997, “Distributed Collaborative Design (DisCollab): An ATP Opportunity,” http://www.mel.nist.gov/msid/groups/edt/ATP/white-paper (Whitepaper of NIST-ATP Workshop: Tools and Technologies for Distributed and Collaborative Design)

[3] Adobe DRM, http://www.adobe.co.uk/epaper/features/drm/drmtools.html

[4] IBM EMMS, http://www-306.ibm.com/software/data/emms/

[5] LockStream, http://www.lockstream.com

[6] Microsoft RMS, http://www.microsoft.com/downloads/details.aspx?FamilyID=be7fae0c-2db2-4f7f-8aa1-416fe1b04fb1&DisplayLang=en

[7] Fetscherin, M., and Schmid, M., 2003, “Comparing the Usage of Digital Rights Management Systems in the Music, Film, and Print Industry,” *ACM Proceedings of the Fifth International Conference on Electronic Commerce*, Pittsburgh, PA, pp. 316–325.

[8] Attrapadung, N., Kobara, K., and Imai, H., 2003, “Broadcast Encryption With Short Keys and Transmissions,” *ACM Proceedings of the 2003 Workshop on Digital Rights Management*, Washington, DC, pp. 55–66.

[9] Adelsbach, A., Katzenbeisser, S., and Veith, H., 2003, “Watermarking Schemes Provably Secure Against Copy and Ambiguity Attacks,” *ACM Proceedings of the 2003 Workshop on Digital Rights Management*, Washington, DC, pp. 111–119.

[10] Chadha, R., Kanovich, M., and Scedrov, A., 2001, “Inductive Methods and Contract-signing Protocols,” *ACM Proceedings of the Eigth Conference on Computer and Communications Security*, Philadelphia, PA, pp. 176–185.

[11] Koenen, R. H., Lacy, J., Mackay, M., and Mitchell, S., 2004, “The Long March to Interoperable Digital Rights Management,” Proc. IEEE, **92**(6), pp. 883–897.

[12] Cox, I. J., Kilian, J., Leightont, T., and Shamoon, T., 1997, “Secure Spread Spectrum Watermarking for Images, Audio and Video,” IEEE Trans. Image Process., **6**(2), pp. 1673–1687.

[13] Barni, M., and Barolini, F., 2004, “Data Hiding for Fighting Piracy,” IEEE Signal Process. Mag., **21**(2), pp. 28–39.

[14] Ohbuchi, R., Masuda, H., and Aono, M., 1998, “Watermarking Three-Dimensional Polygonal Models through Geometric and Topological Modifications,” IEEE J. Sel. Areas Commun., **16**(4), pp. 551–560.

[15] Benedens, O., 1999, “Geometry-Based Watermarking of 3D Models,” IEEE Comput. Graphics Appl., **19**(1), pp. 46–55.

[16] Harte, T., and Bors, A. G., 2002, “Watermarking Graphical Objects,” *IEEE Proceedings of the 14th International Conference on Digital Signal Processing*, Vol. 2, pp. 709–721.

[17] Praun, E., Hoppe, H., and Finkelstein, A., 1999, “Robust Mesh Watermarking,” *ACM Proc. SIGGRAPH’99*, pp. 49–56.

[18] Kanai, S., Date, H., and Kishinami, T., 1998, “Digitial Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition,” *Proceedings of the Sixth IFIP WG5.2/GI International Workshop on Geometric Modelling: Fundamentals & Applications*, Tokyo, pp. 296–307.

[19] Yi, K., Pan, Z., Shi, J., and Zhang, D., 2001, “Robust Mesh Watermarking Based on Multiresolution Processing,” Comput. Graphics, **25**(3), pp. 409–420.

[20] Ohbuchi, R., Masuda, H., and Aono, M., 1998, “Data Embedding Algorithm for Geometrical and Non-geometrical Targets in Three-Dimensional Polygonal Models,” Comput. Commun., **21**, pp. 1344–1354.

[21] Zhang, L., Tong, R., Su, F., and Dong, J., 2002, “A Mesh Watermarking Approach for Appearance Attributes,” *IEEE Proceedings of the Tenth Pacific Conference on Computer Graphics & Applications (PG’02)*, pp. 450–451.

[22] Ohbuchi, R., Masuda, H., and Aono, M., 1999, “A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces,” *IEEE Proceedings of the 1999 Computer Graphics International (CGI’99)*, Canmore, Canada, pp. 180–187.

[23] Lach, J., Mangione-Smith, W. H., and Potkonjak, M., 2001, “Fingerprinting Techniques for Field-Programmable Gate Array Intellectual Property Protection,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Sys., **20**(10), pp. 1253–1261.

[24] Caldwell, A. E., Choi, H.-J., Kahng, A. B., Mantik, S., Potkonjak, M., Qu, G., and Wong, J. L., 2004, “Effective Iterative Techniques for Fingerprinting Design IP,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Sys., **23**(2), pp. 208–215.

[25] Ko, K., Maekawa, T., Patrikalakis, N., Masuda, H., and Wolter, F., 2003, “Shape Intrinsic Fingerprints for Free-Form Object Matching,” *ACM Proceedings of the SM’03*, Seattle, Washington, pp. 196–207.

[26] Sandhu, R. S., and Samarati, P., 1994, “Access Control: Principle and Practice,” IEEE Commun. Mag., **32**(9), pp. 40–48.

[27] Sandhu, R., 1993, “Lattice-Based Access Control Models,” IEEE Comput. Graphics Appl., **26**(11), pp. 9–19.

[28] Sandhu, R., 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.

[29] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R., 2001, “Proposed NIST Standard for Role-Based Access Control,” ACM Trans. Inf. Syst. secur., **4**(3), pp. 224–274.

[30] Georgiadis, C. K., Marvridis, I., Pangalos, G., and Thomas, R. K., 2001, “Flexible Team-Based Access Control Using Contexts,” *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, Chantilly, VA, pp. 21–27.

[31] Thomas, R. K., 1997, “Team-Based Access Control (TMAC): A Primitive for Applying Role-Based Access Controls in Collaborative Environments,” *Proceedings of the Second ACM Workshop on Role-based Access Control*, Fairfax, VA, pp. 13–19.

[32] Harrington, A., and Jensen, C., 2003, “Cryptographic Access Control in a Distributed File System,” *Proceedings of the Eigth ACM Symposium on Access Control Models and Technologies*, Como, Italy, pp. 158–165.

[33] Cera, C., Braude, I., Comer, I., Kim, T., Han, J., and Regli, W., 2003, “Hierarchical Role-Based Viewing for Secure Collaborative CAD,” *Proceedings of the 2003 ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference (DETC/CIE2003)*, Chicago, IL, Paper No. DETC2003/CIE-48277.

[34] Cera, C. D., Kim, T., Han, J. H., and Regli, W. C., 2004, “Role-Based Viewing Envelopes for Information Protection in Collaborative Modeling,” CAD, **36**(9), pp. 873–886.

[35] Wang, Y., Ajoku, P. N., and Nnaji, B. O., 2004, “Scheduled Role-Based Distributed Data Access Control Model for Data Sharing in Collaborative Design,” *Proceedings of the 2004 International Symposium on Collaborative Technologies and Systems (CTS2004)*, San Diego, CA, pp. 191–196.

[36] Wang, Y., Ajoku, P. N., and Nnaji, B. O., 2004, “Distributed Data Access Control for Lean Product Information Sharing in Collaborative Design,” *Proceedings of the 2004 ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference (DETC/CIE2004)*, Salt Lake City, UT, Paper No. DETC2004/CIE-57748.

[37] Kao, K. J., Seeley, C. E., Yin, S., Kolonay, R. M., Rus, T., and Paradis, M., 2003, “Business-to-Business Virtual Collaboration of Aircraft Engine Combustor Design,” *Proceedings of the 2003 ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference (DETC/CIE2003)*, Chicago, IL, Paper No. DETC2003/CIE-48282.

[38] Engineous Software, http://www.engineous.com

[39] Lubell, J., Peak., R. S., Srinivasan, V., and Waterbury, S. C., 2004, “STEP, XML, and UML: Complementary Technologies,” *Proceedings of the 2004 ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference (DETC/CIE2004)*, Salt Lake City, UT, Paper No. DETC2004/CIE-57743.

[40] ISO TC184/SC4/WG11 N223, ISO/WD 10303–28, *Product Data Representation and Exchange: Implementation Methods: XML Schema Governed Representation of EXPRESS Schema Governed Data, 2004-02-17*.

[41] Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org

[42] World Wide Web Consortium, http://www.w3.org

[43] JT Open, http://www.jtopen.com

[44] Web Consortium, http://www.web3d.org

[45] UGS, http://www.ugs.com/products/open/plmxml/

[46] Wang, Y., and Nnaji, B. O., 2004, “UL-PML: Constraint-Enabled Distributed Product Data Model,” Int. J. Prod. Res., **42**(17), pp. 3743–3763.

[47] Lubell, J., and Frechett, S., 2002, “XML Representation of STEP Schemas and Data,” ASME J. Comput. Inf. Sci. Eng., **2**(1), pp. 69–71.

[48] Lubell, J., 2002, “From Model to Markup,” *Proceedings of the 2002 XML Conference*, Baltimore, MD, http://www.mel.nist.gov/msidlibrary/doc/m2m.pdf