The Research of Product Graphical Information Sharing Technology of Virtual Manufacturing Enterprise in E-Commerce Environment

Mingxu Ma

Haiyue Li

Wenlei Zhang

Chaowan Yin

Chengen Wang

The Research of Product Graphical Information Sharing Technology of Virtual Manufacturing Enterprise in E-Commerce Environment

Mingxu Ma [12], Haiyue Li [12], Wenlei Zhang [12], Chaowan Yin [1], Chengen Wang [1]

[1] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100086, China
{ Mingxu MA, Haiyue LI, Wenlei ZHANG , Chaowan YIN, Chengen WANG }@sia.cn

ABSTRACT

This paper has built a product model by UML and corresponding Product Schema. Then we have illuminated transmit mechanism of the product information by a dumbbell XML document. At last, we have pointed out the direction of the research. This research will provide a significative explore to the product data interchange between the members of virtual manufacturing enterprise in e-commerce environment.

Keywords: E-commerce, Product model, UML, XML

1. INTRODUCTION

Along with the rapid development of network technology, the e-commerce technology becomes increasingly ripe. The importance of the e-commerce has been recognized more and more and this kind of trade form has been accepted. Virtual manufacturing enterprise as the important user of e-commerce, comparing with the traditional manufacturing enterprise, has many merits such as sharing infrastructure, outstanding manpower, advanced technology resources, sharing the market risk and researching development cost, sharing market and customer etc. The product data exchange certainly must be carried among the members of virtual manufacturing enterprises in order to share product information. In collaborative design, the visualization technology is one of tools to realize the product data exchange, and can realize the some functions, such as the CAD's documents browsing, marking, reading etc.

At present, many visualization systems (Such as the wiseVIEW of SAMSUNG(SAMSUNG, 2004), the VOLO VIEW EXPRESS of AUTODESK (AUTODESK, 2003), The Browsing Tools of Tsinghua Ying Tai(Ying Tai,2004) etc.) all exist a common question to browse CAD's documents, which is that the documents can be opened by using the corresponding tool only when the documents must have been completely transmitted during document delivery. The more complicated the product increasingly become, the more corresponding huge the product data are also, once completely download will fairly cost time. Sometimes, we needn't to see whole details. When one small part of details of product has been modified, the whole product graphical documents needs to be transmitted again. So the limited network resource is wasted seriously.

2. PRODUCT MODEL WITH UML

Product has many features, such as geometry feature, material feature, manufacturing feature, assembly feature, etc. The each feature has a corresponding class. This paper has built a product model with UML. This model has represented the detailed model structure, especially the geometry feature of product. Shown as the figure 1, the product class is the highest rank class in the model, has its own Product_ ID attribute and Product_ Name attribute. The material class, geometry class, manufacturing class and assembly class are the subclasses of product class. The product geometry feature is core feature and other features, it is very important to comprehensively and felicitously represent the product geometry feature..
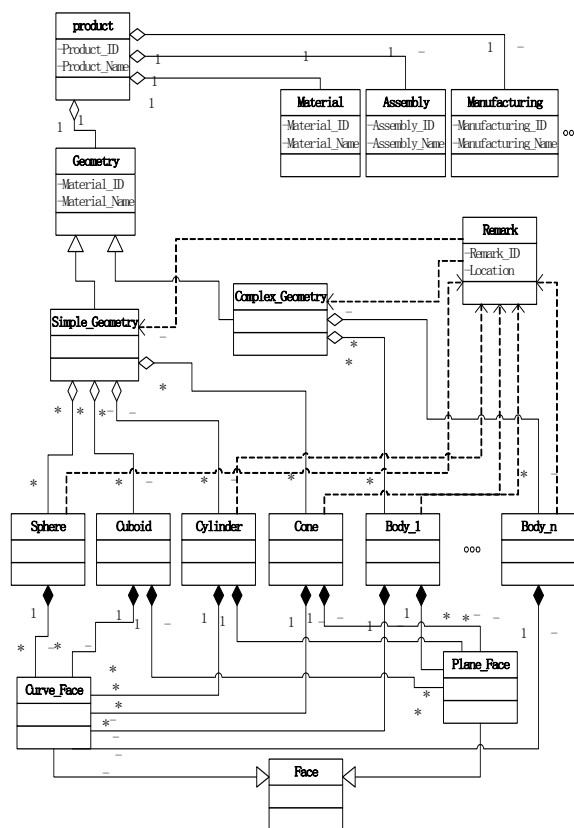

Fig.1 The Product Model

3. THE REPRESENTATION OF PRODUCT DOCUMENT WITH THE SCHEMA OF XML

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.XML has many merits, such as extendability, simpleness and ripeness. So this paper has put forward to a modeling method of the product graphical information based on XML to the virtual enterprises in the e-commerce environment.

Shown as the figure3, according to the Schema of XML(Schema,2003), the paper has described a product data model, including product geometry information and its remark information. Owing to the characteristic of XML itself, this model possesses very strong extensiblity and still can add other product information except mentioned above the geometry information and the remark information, like the material information, the assembly information, the process information and manufacturing information etc.

```xml
<?xml version="1.0" encoding="utf-8" ?>

<xs:schema
targetNamespace="http://localhost:8080/Product.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://localhost:8080/Product.xsd">
    <xs:annotation>
        <xs:documentation>
            This is a product model writed by XML
Schema.
        </xs:documentation>
    </xs:annotation>
    <xs:element name="product_model">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Geometry"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Material"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Assembly"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Manufacturing"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Remark"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="product_ID"
type="xs:ID" use="required"/>
                <xs:attribute name="product_Name"
type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Material"/>
    <xs:element name="Assembly"/>
    <xs:element name="Manufacturing"/>
    <!-- esle product feature -->
    <xs:element name="Geometry">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Simple_Geometry"/>
                <xs:element ref="Complex_Geometry"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Simple_Geometry">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Sphere"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Cuboid"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Cylinder"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Cone" minOccurs="0"
maxOccurs="unbounded"/>
                <!-- esle Geometry feature -->
            </xs:sequence>
            <xs:attribute name="ID" type="xs:ID"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Complex_Geometry">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Entity" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="ID" type="xs:ID"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Remark">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Title"/>
                <xs:element ref="Content"/>
                <xs:element ref="Location"/>
                <xs:element
ref="Correlative_entity_ID"/>
                <!-- esle Remark condition    -->
            </xs:sequence>
            <xs:attribute name="remark_ID"
type="xs:ID" use="required"/>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="Title" type="xs:string"/>
<xs:element name="Content" type="xs:string"/>
<xs:element name="Location"
type="relative_coordinate"/>
<xs:element name="Correlative_entity_ID"/>
<xs:element name="Sphere" type="SphereType"/>
<xs:complexType name="SphereType">
    <xs:all>
        <xs:element ref="Sphere_center"/>
        <xs:element ref="Radius"/>
        <xs:element ref="Color"/>
    </xs:all>
    <xs:attribute name="ID" type="xs:ID"/>
</xs:complexType>
<xs:element name="Sphere_center"
type="relative_coordinate"/>
<xs:element name="Radius" type="xs:double"/>
<xs:element name="Color" type="xs:string"/>
<xs:element name="Cylinder_bottom_center"
type="relative_coordinate"/>
<xs:element name="High" type="xs:double"/>
<xs:element name="Cylinder">
    <xs:complexType>
        <xs:sequence>
            <xs:element
ref="Cylinder_bottom_center"/>
            <xs:element ref="Radius"/>
            <xs:element ref="High"/>
            <xs:element ref="Color"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:ID"/>
    </xs:complexType>
</xs:element>
<xs:element name="Cuboid"/>
<xs:element name="Cone"/>
<xs:element name="Entity"/>
<xs:complexType name="relative_coordinate">
    <xs:sequence>
        <xs:element ref="x"/>
        <xs:element ref="y"/>
        <xs:element ref="z"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="x" type="xs:double"/>
<xs:element name="y" type="xs:double"/>
<xs:element name="z" type="xs:double"/>
</xs:schema>
```

Fig.2 The Product Schema

4. THE EXCHANGE OF PRODUCT INFORMATION

XFI(XFI,2002) allows the exchange of specific information of interest without sending complete XML documents. The most important merit of XFI is to reduce the quantity of communication data and save cost of communication. According to the product Schema having been created in the second paragraph, we have built a simple dumbbell example to show the application of XFI in the data transmission. Then, we have defined the mechanism to create and transmit the product graphical documents and their attached remark. The Figure.3 is a example of a product XML document to describe a dumbbell.

```
<?xml version="1.0" encoding="utf-8"?>


<product_model
xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
nce" xsi:schemaLocation="http://localhost:8080
Product.xsd" xmlns="http://localhost:8080/Product.xsd"
product_ID="P00001">
    <Geometry>
        <Simple_Geometry ID="S00001">
            <Sphere ID="Sp00001">
                <Sphere_center>
                    <x>0</x>
                    <y>0</y>
                    <z>0</z>
                </Sphere_center>
                <Radius>1.0</Radius>
                <Color>red</Color>
            </Sphere>
            <Sphere ID="Sp00002">
                <Sphere_center>
                    <x>4.0</x>
                    <y>0.0</y>
                    <z>0.0</z>
                </Sphere_center>
                <Radius>1.0</Radius>
                <Color>blue</Color>
            </Sphere>
            <Cylinder ID="C00001">
                <Cylinder_bottom_center>
                    <x>0.0</x>
                    <y>0.0</y>
                    <z>0.0</z>
                </Cylinder_bottom_center>
                <Radius>1.0</Radius>
                <High>4.0</High>
                <Color>black</Color>
            </Cylinder>
        </Simple_Geometry>
```

```
    <Complex_Geometry/>
</Geometry>
<Remark remark_ID="R00001">
    <Title>Geometry Remark</Title>
    <Content>the radius is too thick!</Content>
    <Location>
        <x>2.0</x>
        <y>2.0</y>
        <z>2.0</z>
    </Location>

    <Correlative_entity_ID>P00001</Correlative_entity_ID>
    </Remark>
</product_model>
```

Fig.3 The dumbbell XML document according to the Product Schema

4.1 The validity of the fragment in a product XML document

The fragment in a product XML document is a general term to refer to part of an XML document, plus possibly some extra information, that may be useful to use and interchange in the absence of the rest of the XML document. We can appoint the context information of a product graph documents fragment by transmitting the part information rather than the whole product graph documents. So we can achieve the aim to reduce the data transmission quantity.

A valid product XML fragment has a continuous test block according to XML specifications. The figure 4 shows some valid fragments.

a. **0.5**
b. **<Radius>0.5</Radius>**
c. **<Content>the radius is too thick!</Content>**
d. **<Location>**
    <x>2.0</x>
    <y>2.0</y>
    <z>2.0</z>
    </Location>

Fig.4 Some valid product XML fragments

4.2 The transmit mechanism of the product XML fragment

By the most important merit of XFI, we can build a mechanism to realize the aims to reduce the quantity of communication data .

For example, we want to change the radius of the dumbbell' handle , we shouldn't need to transmit whole dumbbell model data before getting to the part of interest. The goal of this paragraph is to define a way to enable processing of small parts of an produt XML document without having to process everything up to the part in question. This can be done regardless of whether the parts are entities or not, and the parts can either be viewed immediately or accumulated for later use, assembly, or other processing.

The following content is the XML fragment document with the right fragment about the change of the dumbbell handle' radius.

```
    <Radius>0.5</Radius>
```

And its context following shows as the figure 5:

```
<?xml version="1.0" encoding="UTF-8"?>
<f:fcs
xmlns:f="http://www.w3.org/2001/XML-fragment"
paraentref="http://localhost:8080/Product.xml"/>
<product_model>
    <Geometry>
        <Simple_Geometry>
            <Cylinder>
                <f:fragbody
fragbodyref="http://localhost:8080/dumbbell_radius
.xml"/>
            </Cylinder>
        </Simple_Geometry>
    </Geometry>
</product_model>
```

Fig.5 The context of XML fragment about the radius of the dumbbell' handle

The context is information to describe the location of the original document fragment. The context include the URL of the original document, the father information of the fragment body, etc. For transmit the fragment information, we can transmit the fragment document and its context to the receiver. The receiver can insert the fragment to the correct location in the original document by the context. In this way, we have realized the transmit mechanism of the product XML fragment.

5. CONCLUSIONS AND FURTHER WORK

This paper has built a product model by UML and corresponding Product Schema. Then we have illuminated transmit mechanism of the product information by a dumbbell XML document. To realize the visualization mechanism, we want to use the Java3D technology. Java3D is a low level 3D scene-graph based graphics programming API for the java language.

Each of the members of virtual manufacturing enterprises in e-commerce can change CAD's documents of oneself into the product graphical document that this paper has built, and show the product graphical document by using the corresponding browsing tool, then achieve the purpose of low cost information sharing owing to the open character of XML. This research will provide a significative explore to the product data interchange between the members of virtual manufacturing enterprise in e-commerce environment.

ACKNOWLEDGEMENT

REFERENCES

[1] WiseVIEW of SAMSUNG. http://www.sds. samsung.com /product/wiseview/wiseview.html. 2004.
[2] VOLO VIEW EXPRESS of AUTODESK. http://usa.autodesk.com/.2003.
[3] Tsinghua Ying Tai. http://www.thit.com.cn/.2004.
[4] Schema of XML. http://www.w3.org/XML/Schema. 2003-4-29.
[5] XFI. http://www.w3.org/TR/xml-fragment. 2002-3-29.