A Web Services Based Platform for Exchange of Procedural CAD Models

Xiang Chen, Min Li, Shuming Gao
State Key Laboratory of CAD&CG, Zhejiang University (310027)
{xchen, limin, smgao}@cad.zju.edu.cn

Abstract

Exchange of procedural CAD models between heterogeneous CAD systems is still a challenging issue in CAD area. Previously we proposed an approach for effectively constructing synchronized collaborative design environment based on heterogeneous CAD systems. In this paper, we extend the synchronized collaborative design environment that we have developed to a web services based platform for exchange of procedural CAD models between heterogeneous CAD systems. First, the real-time exchange of a single operation is extended to the exchange of the complete procedural CAD model between heterogeneous CAD systems. Furthermore, web services technology is adopted to encapsulate the procedural CAD model exchange functions, which are then released on the Internet as a standard interface and can be used by remote developers in their windows applications, web applications, and so on. Finally, a web service for exchange of procedural CAD models between SolidWorks and Autodesk Mechanical Desktop is realized.

Keywords: Procedural CAD model exchange, Heterogeneous CAD systems, Synchronized collaborative design system, Web services.

1. Introduction

In the modern world, the product data exchange is playing its appealing importance along with abundance cooperation carried on between enterprises, because products are often developed via different CAD systems by different enterprise.

The parametric information, such as features, parameters, and constraints in the procedural CAD models, needs to be transferred in the process of the CAD data exchange since it includes the significant design intents which have all along been designers' main concern. The design intents are the functional requirements provided by customers, i.e. a set of geometric and functional rules which the final products have to satisfy. However the exchange of the parametric information between heterogeneous CAD systems is hitherto a tough problem in the field. In search of a feasible way to exchange the information accurately, though, there have been some efforts worldwide. These works have their own characteristics, advantages and also limits, which shall be described in section 2 of our present paper.

We have constructed a synchronized collaborative design platform based on heterogeneous CAD systems recently [1]. In this paper we extend the synchronized collaborative design environment that we have developed to a web services based platform for exchange of procedural CAD models between heterogeneous CAD systems.

2. Related Work

In recent years, the exchange of procedural CAD models between heterogeneous CAD systems has attracted increasingly great attention. Outlined here in the Followings are some of the main works conducted so far.

2.1 ENGEN Project

The ENGEN (Enabling Next GENeration mechanical design) [6] project proposed EDM (ENGEN data model), which is a product data model with parameters, features, design history, and constraints based on STEP Part 42. The purpose of the ENGEN project is to verify exchange capability of design intent, which is represented by design parameters, constraints, and features. The exchange experiments of product information including the design intent are among CAD systems such as Pro/E (PTC), I-DEAS (SDRC), and CADDS5 (CV). But EDM does not have a sufficient number of entities for the design history because it focuses on the exchanging of models with constraints.

2.2 Macro-Parametric Approach

Guk-Heon Choi et al. proposed a macro-parametric approach [3] to exchange CAD model between different CAD systems. By analyzing the general commands of several commercial CAD systems (CATIA, Pro/ENGINEER, UG, IDEADS, SolidWorks, and SolidEdge), they set up a series of neutral commands. The standard commands set is a common set of

modeling commands [4] that are used in part modeling modules of major commercial CAD systems. It does not include commands of assembly or sheet metal modules. The macro-parametric approach does not deal with assembly level such as a mate condition or CAD system's specific and sophisticated commands e.g. the ToroidalBend command of Pro/Engineer. Instead of directly exchanging CAD models, their method exchanges the macro command files between different CAD systems through neutral commands.

This approach is dependent on the macro files of CAD systems (Pro/E can not be achieved in this way because its macro file could not provide sufficient information). And in some cases, internal geometric models are needed to achieve the translation.

2.3 Universal Product Representation Architecture

Besides academic research, there are also some feature-based translators developed by the companies such as ASPire3D [8], Proficiency [9], Theorem [10], TTI [11], etc., the most representative of which would be Collaboration Gateway, the translator developed by Proficiency. In Collaboration Gateway, the Universal Product Representation (UPR) architecture is defined and adopted to provide universal support for all data levels employed by today's CAD systems (It enables an unprecedented level of CAD interoperability through sharing of design intents including features, dimensions, history, assemblies, metal data, and other information). Currently, Collaboration Gateway has been able to support four high-end CAD systems including IDEAS, Pro/ENGINEER, CATIA and Unigraphics.

This product has its fascinating web page but does not provide a standard web service interface for development, which application developers throughout the world are interested in.

2.4 The Latest Extension of the STEP Standard

Some work aiming at extending the international standard ISO 10303 (STEP) [7] to permit the exchange of procedurally defined shape models between CAD systems is being conducted. These types of models are specified by the sequence of operations used to construct the models rather than by the explicit elements they contain, so that they are easy to edit in a receiving system. The specific work includes:

- **ISO 10303-55:** "Procedural and hybrid representation" — is out for ballot as a Draft International Standard;
- **ISO 10303-108:** "Parameterization and constraints for explicit geometric product models"—is in final preparation for publication as an International Standard;
- **ISO 10303-109:** "Kinematic and geometric constraints for assembly models"—is out for ballot as a Draft International Standard;
- **ISO 10303-111:** "Construction history features"—has passed its Committee Draft ballot, and a Draft International Standard version is in preparation.

3. Design of the Procedural CAD Model Exchange Platform Based on Web Services

3.1 Underground procedural CAD model exchange system

We developed a *Synchronized Collaborative Design* (SCD) platform based on heterogeneous CAD systems earlier [1]. It is a replicated architecture with a distinct CAD system at each site which performs product modeling. In each site there are two translators in addition to an independent CAD system. One is the SMO-to-NMC translator which is responsible for translating each SMO just carried out locally into a NMC that will be sent to other sites immediately. Another translator, called NMC-to-SMO translator, is in charge of translating each received NMC from other site into one or more corresponding SMOs of the CAD system. It is these two translators in each site that make possible the real-time exchange of the modeling operations between heterogeneous CAD systems and thus enable the platform to support synchronized collaborative design. In all, the key part of the platform is using Neutral Modeling Commands and CAD systems' APIs to achieve the real-time exchange of modeling operations.

In this work, we first extend the platform to make it capable of supporting procedural CAD model exchange between heterogeneous CAD systems. Fig.1 shows the system structure of the procedural CAD model exchange platform.

Although the core ideas and essential techniques are similar between the synchronized collaborative design platform and the procedural CAD model exchange platform, i.e. using Neutral Modeling Commands and CAD systems' APIs to achieve the exchange of modeling operations between heterogeneous CAD systems, the implementation details are different. In the procedural CAD model exchange platform, since all heterogeneous CAD systems are on the same site, the local NMC sequence is used to achieve the exchange of modeling operations between different CAD systems in this platform. Moreover, since the procedural CAD model exchange platform deals with the whole model rather than a real time modeling operation in the synchronized collaborative design platform, the translators and NMC set need to be modified as follows: the SMOs are extracted by traversing and parsing the

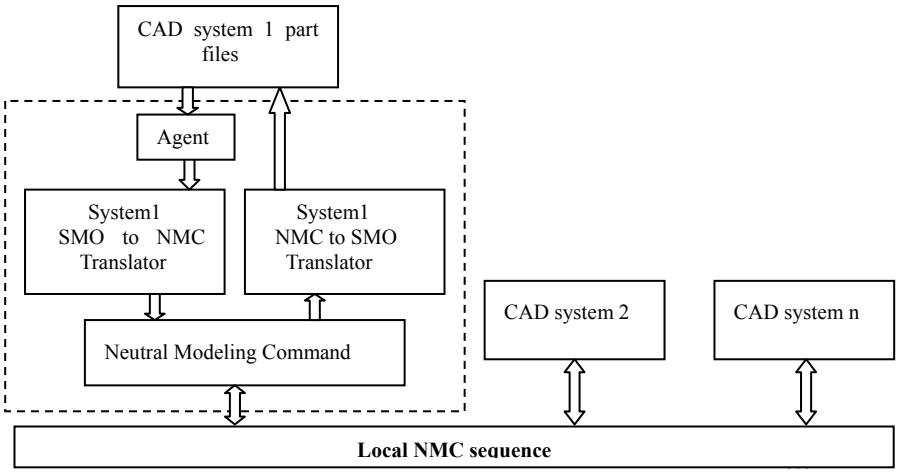

Fig.1. The procedural CAD model exchange platform

feature tree constructed from the part file of the CAD system instead of capturing real time modeling operation event; three new NMCs are added, which will be given in 3.4.

3.2 SOA and Web Services

An SOA (Service-Oriented Architecture) is a component model that inter-relates the different functional units of an application, called services, through well-defined interfaces and contracts between these services. The interface is defined in a neutral manner that should be independent of the hardware platform, the operating system and the programming language in which services are implemented. This allows services, built on a variety of such systems, to interact with each other in a uniform and universal manner.

Web services is a technology that allows applications to communicate with each other in a **platform- and programming language-independent** manner. A Web service is a software interface that describes a collection of operations which can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation for execution or data for exchange with another Web service. A group of Web services interacting together in this manner defines a particular Web service application in a *Service-Oriented Architecture* (SOA).

Web services uses XML that can describe any and all data in a truly platform-independent manner for exchange across systems, thus moving towards loosely-coupled applications. Furthermore, Web services can function on a more abstract level that can reevaluate, modify or handle data types dynamically on demand. So, on a technical level, Web services can handle data much more easily and allow software to communicate more freely.

SOA itself is an abstract concept of how software should be put together. It relies on the more concrete ideas and technologies implemented in XML and Web services, to exist in the software form.

The distinction between SOA services and Web services lies in their designs. The SOA concept does not exactly define how services are specifically to interact, but just how services can understand each other and how they can interact. Web services, on the other hand, has specific guidelines on how messaging between services needs to interact; i.e. the tactical implementation of an SOA model is most commonly seen in SOAP messages delivered over HTTP. Thus, Web services is essentially a specific subset of how an SOA can be implemented.

Through the analysis above, we recognize the

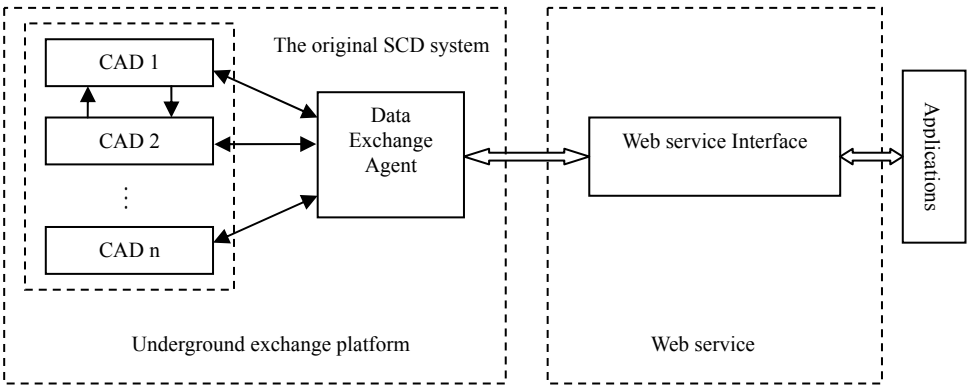

Fig.2. Architecture of procedural CAD model exchange platform based on web services

significance of SOA and decide to develop a web service to encapsulate our procedural CAD model exchange system. Once our web service is published on the internet, any developers are supposed to find this service through UDDI (Universal Description, Discovery and Integration) and use it in their applications or services to exchange their procedural CAD model. In fact, they could simply use it without knowing the details of our procedural CAD model exchange system.

3.3 Architecture of the Web Services Based Platform

Fig.2 shows the architecture of the procedural CAD model exchange platform based on web services**.** The left part is the underground exchange platform, while the right part is the web service module which provides a service interface without revealing the implementing details of the data exchanging.

The Data Exchange Agent module takes charge of dispatching NMCs to a suitable CAD system after the source CAD part file has been uploaded. And then this agent will listen to detect whether the translating work has been done and will send out the transformed CAD part file when assured.

Actually, the application module is not a part of this platform. The user application is any module that needs our service and could use it through the interface we defined. It can be a windows application, a web application, or even an alternative web service, if the developers prefer it.

3.4 Construction Details of the NMC set and the Web Service Interface

The original NMC set used in the synchronized collaborative design platform can not completely satisfy the needs of our procedural CAD model exchange platform. So we extend the NMC set to have three new NMCs: *NewCADFile*, *OpenCADFile* and *SaveCADFile*. Now let us see the specific data flow.

First, when the source CAD part file has been uploaded to the translating system, the data exchange agent will dispatch an OpenCADFile NMC to the suitable CAD system (according to the source CAD part file's type).

Then, in this CAD system, we need to:

1) Open the part file and broadcast a *NewCADFile* NMC to other CAD systems;
2) Traverse the feature tree of the part and extract the SMO sequence and translate each SMO to its corresponding NMC to broadcast;
3) Broadcast a SaveCADFile NMC and CloseCADFile NMC to other CAD systems in the procedural CAD model exchange platform;
4) Close the part file after receiving all acknowledgement messages.

On the other hand, in other CAD systems of the platform, we need to:

1) Create a new part file when receiving a NewCADFile NMC;
2) Translate each modeling NMC to its corresponding SMO;
3) Save the reconstructed part model as a native CAD part file when receiving a *SaveCADFile* NMC;
4) Close the part file when receiving a *CloseCADFile* NMC and send an acknowledgement message back to the CAD system which sends this NMC.

Fig.3 and Fig.4 are the two sample NMC output files (record the NMCs the CAD systems send):

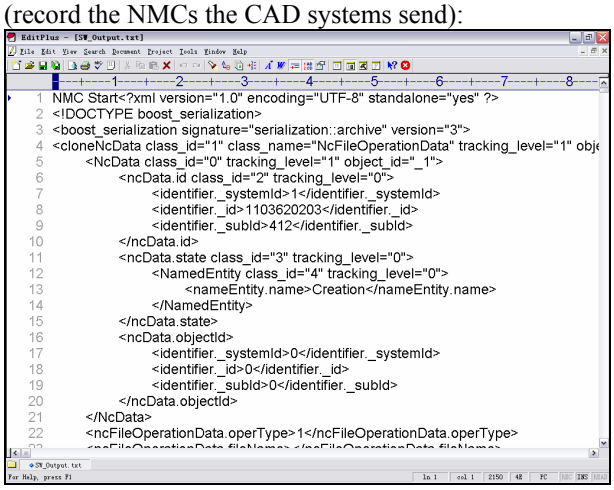

Fig.3. NMC output of the source CAD system

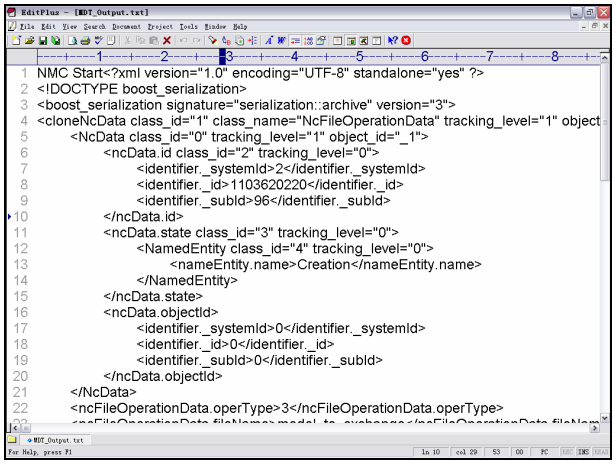

Fig.4. NMC output of the destination CAD system

The web service open interface is as follows:

[WebMethod (Description="send the file to server and get the new file back")]

public string ***TransformFile****(string* ***fileName****, string* ***fileStr****, int* ***srcID****, int* ***desID****)*

```
{
    ...
}
```

The WSDL:

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions
  ...
>
 <types>
  <s:schema elementFormDefault="qualified"
      targetNamespace="http://
      www.cad.zju.edu.cn/">
   <s:element name="SendFile">
    <s:complexType>
     <s:sequence>
       <s:element minOccurs="0"
            maxOccurs="1" name="fileName"
            type="s:string" />
       <s:element minOccurs="0"
            maxOccurs="1" name="fileStr"
            type="s:string" />
       <s:element minOccurs="1"
             maxOccurs="1" name="srcID"
             type="s:int" />
       <s:element minOccurs="1"
            maxOccurs="1" name="desID"
            type="s:int" />
     </s:sequence>
    </s:complexType>
   </s:element>
   <s:element name="SendFileResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0"
              maxOccurs="1"
              name="SendFileResult"
              type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
  </types>
```

Parameters:

fileName: the name of the source CAD part file (e.g. a MDT file named example_partfile.dwg).

fileStr: the content of the source CAD part file (in base64string format).

srcID: an integer number representing the type of the source CAD part file.

desID: an integer number representing the type of the destination CAD part file.

Return:

The content of the destination CAD part file (also in base64string format).

Function:

Take in the parameters and transform the source CAD part file to a new CAD part file (the user defines the new file's type in parameter desID), then return it back.

The data exchanging between the web service and the underground system is achieved in TCP/IP protocol as we take into account its high efficiency.

4. Implementation

A web service for exchange of procedural CAD models between SolidWorks and Autodesk Mechanical Desktop is realized. For each of the two CAD systems, both SMO-to-NMC and NMC-to-SMO translators are implemented with Visual C++ 6.0 and the open programming APIs of the CAD systems (SolidWorks 2003 and MDT6.0). The web-service part of the platform is implemented with Microsoft C#. And a web application is also developed to invoke the web service with a view to checking whether it works well. It is implemented with Microsoft C# too.

Now let us have a look at how all these modules work together appropriately.

First, the user uploads a CAD part file (push the upload button) on the web end (Fig.5) and pushes the suitable transform button to get the specific CAD file that he wants (e.g. push the *Get_MDTFile* button to get the transformed MDT file).

When a transform button is pushed down, the web service is invoked and then it starts the underground exchange platform (the communication between web service module and the underground exchange platform is built on TCP/IP). Next, the underground exchange platform produces a new CAD file and transfers it back to the web service module.

At last, the web end gets the transformed CAD part file returned from the web service (Fig.6).

The SolidWorks before transforming and the MDT part file after transforming are respectively shown in Fig.7 and Fig.8.

5. Conclusion and Future Work

In this paper, we present a web services based platform for exchange of procedural CAD models between heterogeneous CAD systems. The features of the platform are:

1) The exchange of procedural CAD models between heterogeneous CAD systems is achieved based on Neutral Modeling Commands and the APIs of CAD systems.
2) Web services technique is used to construct a standard interface for the procedural CAD model exchange platform so that it can be used by remote developers in their windows applications, web applications, etc.

In the future, we will add more CAD systems such as Pro/ENGINEER, UG, etc. into this platform and adjust the web-service structure to provide more powerful functions.

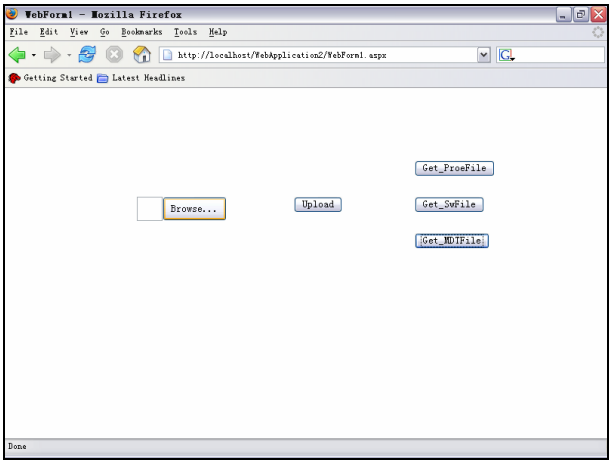

Fig.5. View of the web application

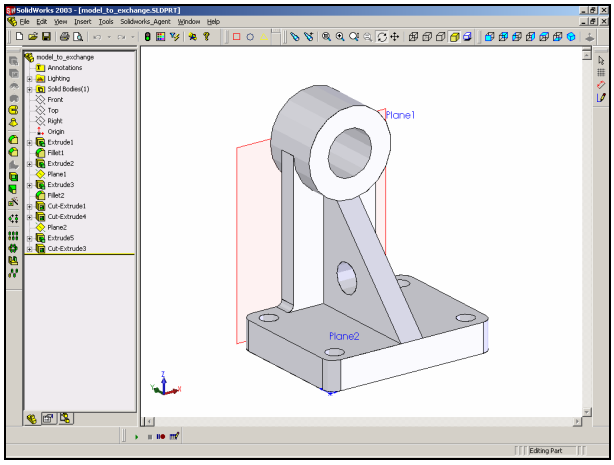

Fig.7. View of the source SolidWorks part file

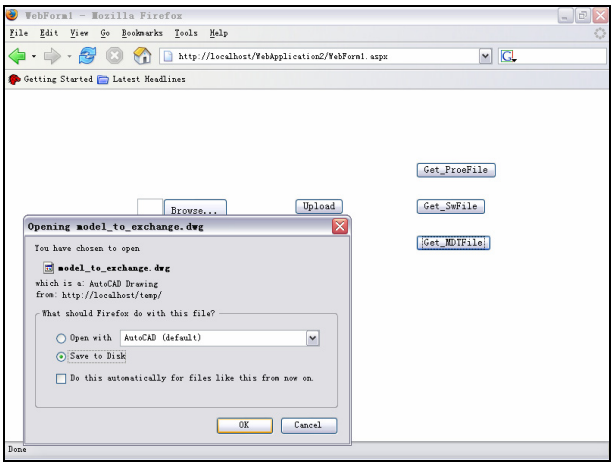

Fig.6. Get the transformed MDT part file

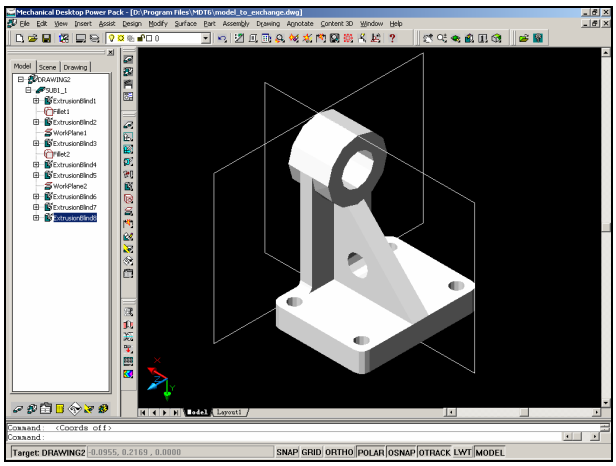

Fig.8. View of the destination MDT part file

Acknowledgments

The authors feel greatly indebted to the financial support from NSF of China (No.60273057) and the Trans-Century Training Programme Foundation for Talents by the Education Ministry of China.

References

[1] Min Li, Shuming Gao, Jie Li, Youdong Yang, "An approach to supporting synchronized collaborative design within heterogeneous CAD systems", *Proceedings, ASME CIE/DETC*, 2004.

[2] Min Li, Youdong Yang, Jie Li, Shuming Gao, "A preliminary study on synchronized collaborative design based on heterogeneous CAD systems", Proceedings, CSCWD 2004, 8th *International Conference on Computer Supported Cooperative Work in Design*, May 2004, Xiamen, China

[3] Guk-Heon Choi, Duhwan Mun, Soonhung Han. "Exchange of CAD part models based on the macro-parametric approach", *International Journal of CAD/CAM* 2(2002), 23-31.

[4] Duhwan Mun, Soonhung Han, Junhwan Kim, Youchon Oh. "A set of standard modeling commands for the history-based parametric approach", *Computer-Aided Design* 35 (2003) 1171-1179.

[5] Ari Rappoport. "An Architecture for Universal CAD Data Exchange", Proceedings, *Solid Modeling* '03, June 2003, Seattle, Washington, ACM Press.

[6] Anderson B. "ENGEN data model: a neutral model to capture design intent", *PROLAMAT*98; 1998.

[7] Michael J. Pratt. "Extension of ISO 10303, the STEP Standard, for the Exchange of Procedural Shape Models", *International Conference on Shape Modeling and Applications* 2004, Genova, Italy.

[8] ASPire3D. http://www.aspire3d.com.

[9] Proficiency. http://www.proficiency.com.

[10] Theorem. http://www.theorem.co.uk.

[11] TTI. http://www.translationtech.com.

[12] IBM. http://www.ibm.com/developerworks.