A web-based collaborative framework for facilitating decision making on a 3D design developing process

Purevdorj Nyamsuren\textsuperscript{a,b}, Soo-Hong Lee\textsuperscript{a,\textsuperscript{n}}, Hyun-Tae Hwang\textsuperscript{a}, Tae-Joo Kim\textsuperscript{a}

\textsuperscript{a}Department of Mechanical Engineering, Yonsei University, 50, Yonsei-ro, Seodaemun-gu, Seoul, Korea
\textsuperscript{b}Department of Technical Mechanics, Mongolian University of Science and Technology, 8 horoo, Bagh-toiruu, Sukhbaatar district, Ulaambaatar, Mongolia

Received 28 December 2014; received in revised form 22 February 2015; accepted 23 February 2015
Available online 18 March 2015

Abstract

Increased competitive challenges are forcing companies to find better ways to bring their applications to market faster. Distributed development environments can help companies improve their time-to-market by enabling parallel activities. Although, such environments still have their limitations in real-time communication and real-time collaboration during the product development process. This paper describes a web-based collaborative framework which has been developed to support the decision making on a 3D design developing process. The paper describes 3D design file for the discussion that contains all relevant annotations on its surface and their visualization on the user interface for design changing. The framework includes a native CAD data converting module, 3D data based real-time communication module, revision control module for 3D data and some sub-modules such as data storage and data management. We also discuss some raised issues in the project and the steps underway to address them.

© 2015 Society of CAD/CAM Engineers. Production and hosting by Elsevier. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Decision making; Distributed environment; 3D data visualization; Revision control; WebGL

1. Introduction

Traditional computer-aided design (CAD) and product lifecycle management (PLM) tools have their limitations in supporting quick-to-market, collaborative product development environment [1]. Nowadays, product development is the result of a network-based collaborative process, because most of the projects require co-operation among geographically distributed expert groups with diverse competence [2]. In a distributed developing process, effective and high speed communication is needed to face challenges like short time-to-market and high cost efficiency. Most of the design errors are due to lack of communication between the distributed design teams and manufacturing experts.

Distributed information system frameworks that have been applied to collaborative product design systems can be classified into three main approaches: web services, remote services, and remote repositories [3]. The web service-based frameworks have advantages over remote service and remote repository-based frameworks, and they are relatively simple to design, implement, reduce in software installation costs and capability of collaborative work [3,4].

Various types of web-service based PLM applications have been used in industry [5]. Essentially, such applications only manage information that can be stored in the database tables, or they manage different types of drawings and bills of materials [6]. Lack of integration of product lifecycle management and collaboration tools leads to an increasing problem in large industrial project.

Decision making process on a 3D design developing through the web-based collaborative framework uses informal communication forms such as real-time chats and video conference. However, there are some drawbacks for visualization of the design change and design errors on a decision making process using a web technology. Also, there is a possible limitation to mark the errors on the surface of 3D...
model and give alert by annotation to collaborative designers. These drawbacks cause complications to the collaborative work, and decision making time takes longer.

This work introduces a new web-based framework for facilitating decision making on a 3D design developing process. Such framework contains following functions: annotation on the surface of 3D data, revision control between the 3D designs, real-time collaboration on the text and 2D image document, text chatting and video conferencing.

The structure of this paper is as follows; Section 2 summarizes the architecture of the framework and implemented technique, Section 3 demonstrates framework functions, and Section 4 demonstrates case studies. Conclusions and future contributions are listed in Section 5.

2. System architecture

2.1. Overall structure

Fig. 1 illustrates structure of a collaborative framework for facilitating decision making on a 3D design process.

The framework is consists of: (1) a 3D design module, (2) data management module, and (3) real-time communication module.

The design server contains a STL converter API (Application Programming Interface) of the commercial CAD programs and converter of the 3D Java object. Such functions as uploading and downloading files, authenticating users, revising data and accessing data storage are managed by using MySQL database and PHP server through the data management module. Also, 3D design and visualization related JavaScripts codes are located on the main web-server. Client side java applets consists of a Web-based 3D file visualizing function, difference visualizing function and 3D annotation function. That functions are generated via a JavaScript based WebGL technology.

Real-time communication module contains a text chatting server and a video conference server. Such servers as video and text chatting are generated via open source named Node.js. The advantage of separating activities of this real-time communication module is that one system can continue working regardless the breakage of the sustainable activity of the other one due to overloads.

Real-time communication module contains a text chatting server and a video conference server. Such servers as video and text chatting are generated via open source named Node.js. The advantage of separating activities of this real-time communication module is that one system can continue working regardless the breakage of the sustainable activity of the other one due to overloads.

In the most of PLM systems, users are able to see information for changing versions of repaired design data files as “text data type”. By using this web-based framework it is possible to make 3D visualization of the product geometry change, also to correct neccessary section of 3D data by making annotation and show real-time to collaborative users. The users communicate with each other using text chatting and video conferencing features. Decision making related files on 3D model are stored on data storage.

Fig. 1. A Structure of the web-based decision making system.
Furthermore, all comments and notes are stored with the document history where the comment was added.

2.2. Significance of implemented techniques

Web technology has been widely used to facilitate interoperability between different hardware and software solutions, and between different computer architectures and API. Consequently, most of the distributed and collaborative design systems have been developed with the web technology.

For the proposed system, web technologies, namely JavaScript based WebGL and C are an ideal platform for 3D annotation and 3D differences visualization module. It enables the creation of platform-independent software tools, and also its prepared availability enables users to load Java-based objects and link them to create full applications. Microsoft Visual C is suitable for developing APIs to support a solid and geometric modeling.

For the real-time communication function, we use an open source Node.js which is written in JavaScript. Node.js is a software platform that is used to build scalable network (especially server-side) applications.

Consequently, PHP and MySQL are chosen basic technologies to develop a web-based collaborative framework.

This system helps to increase the speed of communication of decision making on a 3D CAD design developing in the execution and management of projects, and completely isolates the necessity for software updates, installed 3D or CAD viewers and real-time communication tool or licensing.

3. Framework components

The framework contains following functions: (1) notification on the surface of 3D data, (2) revision control between the 3D design files, (3) real-time collaboration through the 2D image document, (4) text chatting and (5) video conferencing.

3.1. Conversion of 3D CAD data

Recently, several CAD programs are used to develop products. For example, CATIA is used to design car bodies and Pro/Engineer is mainly used for designing powertrain. Even though CAD software can exchange native files with each other, they cannot control differences between 3D models designed by different softwares. Therefore, a conversion server that transforms native files created by various CAD systems into neutral format like STL is required in order to resolve the problem. Also the STL file enables to make the visualization of 3D model on the web browser.

Conversion program is built on C. Fig. 2 illustrates the structure of a conversion. The system which converts the design file created by users into STL format works simultaneously with the system installed on the 3D design module and the both systems exchange data through FTP (File Transfer Protocol).

The STL converter works alone while others require both of the CAD packages installed on the server as they use code (API) from this system to write the data.

3.2. 3D CAD data revision control

In the most of PLM systems, users are able to see information for changing versions of repaired design data files as “text data” type [7]. It is a time-consuming process because each information included in the version of geometry data is visualized separately on each interface. To resolve such an inconvenience, comparing design data files during repairing process and viewing changes of shape as in 3D are necessary. Furthermore, most of the 3D CAD programs have functions to recognize and show differences of design data files in 3D, but PLM programs and collaborative design environment do not have any similar function.

To determine shape differences between 3D geometry data, most of the CAD programs use a Spatial Occupancy Comparison Method studied by Brière-Côté [8], which is also used in the suggested system. In 2011, Evan Wallace introduced a new WebGL based CSG.js [9] open source for the calculation of the differences between simple 3D shapes such as cube, cylinder and sphere. Fig. 3 shows basic Boolean operations. A set of three operations (union, subtract, intersect) allows to calculate differences between two CAD models.

The steps of a difference identification process used by CSG (Constructive Solid Geometry) operation are shown in Fig. 4. As mentioned previously, the main point of this methodology is an intersection of two shapes (A, B) as a complex geometric

Fig. 2. A flow chart of STL conversion program.
object. Multiple versioned files could be visualized by highlighting added or removed geometry with colors. It is necessary to convert 3D models created by commercial CAD software into neutral 3D files (such as STL, STEP etc.) in order to visualize them on the website. The STL geometry data is supposed to be converted into the Array form in JavaScript for visualizing the differences between 3D models by using Boolean operation.

Fig. 5 shows structure of the JS (JavaScript) design data file for the difference visualization. This 3D JavaScript file used for calculation of the design changes.

The algorithm is applied only if the design has a history of more than two. The added and removed shape of the reference version (A) compared to the modified version of (B) represented by blue and red colors respectively.

Most of the CAD difference viewers may find changes from the original model to updated model, but they are not able to compare changes between updated models because original model is the only file to be considered as a reference model. Nonetheless, the suggested system allows users to define the reference version of the model and the compared version of the model.

3.3. 3D data based annotation and comment

3D mediated communication is nowadays used in manufacturing industry product lifecycle management. In 2013, Siltanen [10] proposed simple method for 3D mediated collaboration in product lifecycle management and developed web-based 3D mediated collaboration framework. 3D mediated collaboration as such gives a benefit of letting remote users to interact with the same 3D model simultaneously share the ideas about the problem at hand.

Nevertheless, the 3D model can also be used for linking discussions to the context of the discussed product component and design related discussions are to be written on the other section of a web-browser in text type.

It means, precisely describing about which section of the 3D model needs a correction in a text form is necessary. In such frameworks, the 3D models are connected with the user only by interactions like zoom in, zoom out, and rotate [11].

But in case of our proposed system, besides the previous interaction (rotate, zoom in, zoom out) the 3D model is used interactively in the process of decision making. In other words, users can comment-write annotations on any desired section or surface of the 3D model. A special sphere-shaped-mark will be created on the annotation and comment written parts of the 3D model and other users can view it easily by mouse over action.
Fig. 6 shows the 3D model which is used for a discussion process. Overall 3 different users take part in discussion and each creates an annotation on the 3D model surface. The architecture of annotation writing and process of sharing to other customers are shown in Fig. 7. HTML5 and JavaScript based WebGL technology enable the 3D model visualization and interaction with 3D model.

WebGL (Web-based Graphics Language) is an effective technology to visualize 3D graphics on a web browser. It is a JavaScript API that interacts with the Graphics Processing Unit (GPU). Diverse WebGL frameworks are available for 3D visualization and product development processes such as OSG, Oak3D, J3D, X3DOM, and Three.js.

Three.js, a JavaScript library provided to draw WebGL-based 3D model on the web interface, is one of the most widely used frameworks. While hundred lines of a JavaScript code are required to construct a simple object in raw WebGL, only a few lines are required to construct the same object within a few lines in Three.js.

Left side of Fig. 7 shows JavaScript related functions like 3D visualization, 3D annotation and 3D revision control. Above mentioned JavaScript library functions are shown in Table 1.

### 3.4. Real-time communication on decision making

Since mid-1990s, parametric CAD/CAM/CAE systems are integrated with other PLM components, and application of PLM systems is increased in the global market [12,13]. From that era, various communication systems have been employed in product development and it shortens production time and reduces product cost. Therefore, distributed teams are faced to use real-time communication tools for collaborative work.

In a global manufacturing market, it is more important than ever to have up-to-date, real-time communication in distributed development and share information with clients worldwide. Real-time communication (RTC) is any modes of telecommunication in which all users can exchange information instantly or with negligible latency. Following variations of real time communication are used in collaborative design development:

1. IRC (internet relay chat) or other chatting modes.
2. Live video conference communications.

These communication variations are all held by a browser-to-browser message delivering process.

There are several services for delivering browser to browser messages, such as Google App engine and Adobe Livecycle Collaboration Service. The most promising new message delivery method seems to be WebSocket [14], which provides bi-directional communications channels over TSP. WebSocket protocol is being standardized by the IETF, and the application programming interface, the WebSocket API [15], is being standardized by the W3C. WebSocket protocol is not supported by all browser yet, mainly because of security issues addressed in the first protocol standard versions.

The JavaScript based Node.js framework developed by Google that is based on WebSocket protocol is the most promising open source. Node.js is a software platform that is used to build scalable network (especially server-side) applications [16]. Node.js utilizes JavaScript as its scripting language, and achieves high throughput via non-blocking I/O and a single-threaded event loop. Node.js contains a built-in HTTP server library, making it possible to run a web server without the use of external software, such as Apache or Lighttpd, and allowing more control of how the web server works.

The major difference between Node and other server-side technologies is Node’s use of a single thread and asynchronous architecture. Many other server-side technologies are multi-threaded and synchronous, meaning that threads can be blocked while waiting for replies from the database. Each request creates a new thread from a limited pool based on system RAM usage. Node’s asynchronous design allows it to handle a large number of concurrent connections with high throughput on a single-thread, which makes it highly scalable.

Node is not meant as a replacement for other technology stacks, but it can provide scalability and increased performance to applications which fits its purpose. Some examples of

---

**Table 1: Functions of the JavaScript.**

<table>
<thead>
<tr>
<th></th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Three.js is a lightweight cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a Web browser</td>
</tr>
<tr>
<td>2</td>
<td>CSG.js is a modeling technique that uses Boolean operations like union and intersection to combine 3D solids</td>
</tr>
<tr>
<td>3</td>
<td>STLLoader.js converts STL files into the WebGL based 3D files</td>
</tr>
<tr>
<td>4</td>
<td>Detector.js determines whether a current web browser supports WebGL or not</td>
</tr>
<tr>
<td>5</td>
<td>jQuery.js is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.</td>
</tr>
<tr>
<td>6</td>
<td>Html2canvas.js translates 3D design files to the 2D image file.</td>
</tr>
</tbody>
</table>
application types which can benefit from using Node are REST APIs, Chat applications and Real-Time Tracking applications (Brokerage trading dashboards, real-time user statistics, etc.)

Fig. 8 shows the structure of node.js based real-time communication module. Besides a video conference, and text messaging functions, 2D image based real-time annotation functions complete the real-time communication module.

These functions are all installed on different servers and information or data sent from any client browser is transferred to all users vice versa becoming steam.

4. Implementation of a framework

Fig. 9 shows main user interface of a suggested framework. User interface is built using main web technologies PHP, HTML5, MySQL and APIs provided by two external libraries: Three.js for visualizing and interacting 3D models and Node.js for real time communication on a decision making process.

The STL models are loaded on the web browser through the JavaScript library. Each user can see the same model and interact with it by selecting 3D model components putting comments and annotations on the surface of 3D model. Of course user interface has a basic functions of rotating the model for viewing from different angles. Annotations entered by users appear like different colored spheres on the 3D model surface. This makes it possible to view easily what part of the 3D model has been commented by which user. Fig. 9 shows that overall 3 users participate in the 3D discussion process.

At left side of User Interface, instant messages linked to the 3D model under discussion. However, when annotation is put on the surface of 3D model, related information is automatically updated into Related Annotations part of the User interface. Additionally, 3D model linked and discussion related files are shown on the right side of web-browser.

Fig. 10 shows the parts to be discussed about the picture on white board. Each time the user interacts with the model, the transformation matrix defining the change of model viewing angle is sent to the other users' browser using WebSockets and the viewing angle in the external browser is changed accordingly.

Fig. 11 demonstrates the case study about the CAD model with two different versions. Fig. 11(a) and (b) shows the reference and updated models, modeled by Solidworks2010.
Fig. 10. A window of whiteboard.

Fig. 11. A window of revision visualization.

Discussion on 3D design changes
The two significant differences between these models are the cutted hole on the left surface and the existence of a horizontal hole on the surface, which is only visible in an updated model.

In Fig. 11(c) red and blue colors imply deleted and added section of the model, respectively.

Table 2 shows a comparison of the commercial system and the system that was developed in this paper. All the compared systems are supporting the web-based 3D model collaboration. Core technology of this system is make a comment on the surface of the model, and the comment point is to be updated rapidly to other users in real time. Using Node.js and WebSocket, the system provides sharing the coordinate data of the annotation point but view angle is not be shared. The system is similar to other document sharing systems (eg Google Docs) that sharing a position of cursor not the view of users. It can give freedom to user with display. Compared to the commercial system, it includes a comprehensive set of all the features of the commercial system, and the system provides the real-time capabilities with the annotation on surface via Node.js and WebSocket.

The two significant differences between these models are the cutted hole on the left surface and the existence of a horizontal hole on the surface, which is only visible in an updated model. In Fig. 11(c) red and blue colors imply deleted and added section of the model, respectively.

Fig. 12 shows file storage of this framework. When a new discussion starts on the 3D model, the system automatically creates a discussion related folder. While the discussion processes are all being made, actions are saved in time schedule part. By using the time schedule window, users can find out the date which the project update was made.

5. Conclusions and future works

Due to the integration of distributed resources such as human beings, engineering tools, and large variety of product related information, a product designing process in a 3D environment requires lots of efforts. Moreover, the PLM system for decision making on the 3D designing process has been shifting toward to a web-based system to provide a distributed design environment. Especially, discussion on the 3D designing process and visualization of design changes on
the web-browser are very complicated, and lots of resources are needed to be allocated as a result.

The web-based decision making system can help to reduce those requirements significantly. WebGL is a main tool for visualizing an interactive 3D model under the designing process on a web-browser. Also, the JavaScript based software platform for communicating tool is applied on the server side.

Yet, the suggested system is only applicable for discussion on the simple 3D model designing process. The system should be developed further to support more complicated 3D models with special forms; moreover, a real time communication tool among users in different environments should be supported as well in the future.

References


