Robotics and Computer-Integrated Manufacturing 26 (2010) 778–789

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

A distributed and interactive system to integrated design and simulation for collaborative product development

Hongwei Wang [a,b], Heming Zhang [a,*]

[a] National CIMS Engineering Research Centre, Tsinghua University, Department of Automation, Beijing 100084, PR China
[b] Engineering Design Centre, Cambridge University Engineering Department, Cambridge CB2 1PZ, UK

ARTICLE INFO

Article history:
Received 26 April 2008
Received in revised form
20 January 2010
Accepted 11 May 2010

Keywords:
Collaborative product development
Integrated design
Distributed simulation
Engineering design
Web Services

ABSTRACT

The goal of applying collaborative product development in industry has raised the need to develop software tools supporting system integration and group collaboration. Current methods and tools mainly focus on the collaborative creation of design components and assemblies. However, few of them support the collaborative work in developing simulation models so that proposed design concepts and solutions can be evaluated by integrating expertise from several disciplines. The purpose of this research is to develop a distributed and interactive system on which designers and experts can work together to create, integrate and run simulations for engineering design. To develop such a system, a number of issues, e.g. effectiveness and efficiency of modeling work, the re-use of models, interaction and cooperation, accuracy of simulation, collaborative operation on models, etc., need to be addressed. This paper describes an open architecture to developing simulations for engineering design in a distributed and collaborative environment, identifies a set of key issues raised in this architecture, and presents the techniques employed in our solution.

1. Introduction

Engineering design involves the application of technical knowledge in a structured way to produce the definition of a product that meets specified needs. The industrial scene continues to change and modern design engineers are subject to demanding pressures, e.g. rapidly changing technology, increasingly complex technical systems, and working in large, multidisciplinary teams [1]. Two strategic ways have been proposed and researched to help designers make better decisions throughout the design process, namely virtual reality simulation and rapid prototyping [1]. The former method is also referred as Virtual Prototyping (VP). One of the important advantages of VP models is their digital nature which, coupled with much faster and affordable computer processing power, permits revision and optimization of the functionality of the designed parts in a very fast, economic and efficient manner [2].

Recently, another solution called Collaborative Product Development (CPD) is also widely studied by the research community to improve the decision-making of designer engineers. A distributed CPD system is particularly helpful for modern product development which is being done more often by geographically and temporally distributed design teams [3]. A number of issues have been taken into account in the development of CPD systems, e.g. information system architecture, communication tools, engineering applications, product geometric representation, etc. [4]. Existent CPD systems mainly focus on supporting such activities as common access of data and collaborative visualization and design of components and assemblies [4]. However, few of these tools support the collaborative work in modeling and simulation to forecast the performance of proposed concepts and solutions. By applying Collaborative Modeling and Simulation (CMS) in a design process, combined advantages of both VP and CPD can be achieved. Moreover, other issues can also be considered in an early design stage, e.g. manufacturability, design of components belonging to other disciplines, etc.

The research questions raised in this research are; what kind of simulations are particular suitable for applying CMS; what system architecture the CMS system in a distributed environment should have; what technologies should be used to support system integration and group collaboration; how design engineers and domain experts can cooperate on solving a problem; how to improve the effectiveness, efficiency, and accuracy of simulations. The developments in related domains can offer some help. First, commercial Computer-Aided Engineering (CAE) tools have been widely applied in industry and computer programs for specific analysis tasks have also been developed. Secondly, with the rapid advancement of information and communication technologies,

* Corresponding author.
E-mail addresses: hmz@mail.tsinghua.edu.cn, batilover@gmail.com (H. Zhang).

0736-5845/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2010.05.005

systems integration and collaboration technologies have been developed and deployed in different engineering application domains [5]. These put two further questions forward: how to support the integration of tools and how to re-use the models and codes created in previous designs.

To answer these questions, we have proposed a Web-based solution and developed a prototype CMS system. The development of a CMS system actually involves a wide range of research topics and some of them have been published elsewhere, e.g. the model-driven modeling approach and the distributed interaction mechanisms between simulation models [6,7]. In the remainder of this paper, we will present a scenario of applying the CMS system in a design process, system architecture of the CMS system, key issues raised in the architecture and our solutions.

2. Related work

CPD is a methodology of integrated and collaborative design for addressing the requirements raised by the increasingly complex nature of modern product development [8,9]. Therefore, system integration and collaboration can be identified as the key issue of CPD research, which has been researched widely [10,11]. In the review of Li et al., [10] two types of collaborations are identified, horizontal collaboration and hierarchical collaboration. The former concerns the cooperation of team members from the same discipline to carry out a complex task in either a synchronous or asynchronous way while the latter emphasizes the cooperation between upstream design and downstream manufacturing [10]. A number of methods have been developed in terms of how several systems, or components within a system, can be integrated, namely, agent-based integration, Web-based integration and the integration of the two [11]. It is noteworthy that a single technology, e.g. agent or Web, can actually support both integration and collaboration. Therefore, we do not use integration and collaboration to classify the related works but instead focus on its key enabling technologies, namely information representation and sharing, agent-based collaborative design, infrastructural technologies and system frameworks, distributed modeling and simulation.

Information representation and sharing is fundamental to CPD as it involves the objects and their relationship in a design solution. In collaborative design, information for visualization, products, project management, etc. needs to be represented and shared [10]. The ultimate goal of representing and sharing product information is to make it transferred to, and accessed by authorized users or computer systems. Szykman et al., [12] proposed a product model to make design information assessable to users in a Web-based system. To make the shared information more easily interpreted, Kim et al., [13] developed an ontology-based assembly design method which aims to make heterogeneous modeling terms semantically processed both by design collaborators and intelligent systems. To achieve optimized transferring of information, 3D streaming technology has also been studied [10]. Chu et al., [14] implemented a 3D design environment where the information with different levels of detail is transferred to different design engineers.

The advantages of agent-based approach includes: proactive object systems and simplification of system architecture; desired modularity of system and improved software reusability; and effective and intelligent reaction to ill-structured and dynamic situations [5]. It has been applied in intelligent manufacturing, enterprise integration and supply chain management [15–17]. Wang et al., [18] proposed a collaborative design system which employs several software agents to interact with each other and perform such tasks as communication, product data management, etc. Hao et al., [19] developed a collaborative e-Engineering environment on which an industrial case was studied. Chao et al., [20] proposed an agent-based approach to engineering design which aims to use the agent attributes such as proactiveness and autonomy to achieve effective integration of design tools. To obtain improved capabilities, agent technology has also been integrated with other technologies [11]. For example, Wang et al., [21] developed a Web/agent-based multidisciplinary design optimization environment and Shen et al., [22] proposed an agent-based service-oriented integration architecture for collaborative intelligent manufacturing.

Interoperability between computer-aided engineering software tools is of significant importance to achieve greatly increased benefits in a new generation of product development systems [3]. Lots of work has been done on the system integration and collaboration technologies, e.g. Web, agent, Web Services and semantic Web, etc., and open standards and commercial tools have already been available [5]. The development of infrastructural technology for CPD systems has also been researched to support distributed integration and collaboration with improved effectiveness and efficiency. Fan et al., [23] developed a distributed collaborative design framework using Peer to Peer (P2P) technology and grid technology. Cheng and Fen [24] developed a Web-based distributed problem-solving environment where computational codes can be accessed and integrated to solve engineering problems. Simulation technology is playing an increasingly important role in design validation and verification [25], which raises the need to develop infrastructure for distributed modeling and simulation in the context of CPD.

Reed et al., [26] developed a Web-based modeling and simulation system which was applied to the aircraft design process and argued that such a system could improve the design process. Byrne et al. [27] reviewed recent research on Web-based Simulation (WBS) and its supporting tools and concluded a number of advantages of WBS, including: easy use; collaboration features; licence and deployment models, etc.. One of the important enabling technologies of WBS is the middleware which enables different modules in a WBS to interoperate [27]. Some middleware technologies, e.g. CORBA, Web Services and the High Level Architecture (HLA), have been used in developing collaborative simulation systems for engineering design [8,28,29]. Among these technologies, Web Services is a very promising technology for system integration on the Web whilst HLA is an important and heavily researched standard for distributed simulation [27]. Two ways have been identified and researched for the integrated use of Web Services and HLA, namely developing HLA enabling tools using Web Services and making HLA federation to interoperate with other software applications [30–32].

In summary, lots of methods and infrastructural technologies have been developed and applied to support system integration and collaboration for CPD. The capability of Web-based system is to enable collaborative work in an Internet-distributed environment. Web Services technology offers a flexible integration framework where more dynamic interoperations between modules of a system can be supported. HLA is particularly powerful for applications requiring complex interaction, coordination and synchronization. To develop a distributed modeling and simulation system on which designers and experts can work together to create, integrate and run simulations, an integrated framework based on these technologies needs to be developed. Using Web and Web Services can achieve the goal of supporting collaborative work and dynamic integration of simulations. However, new requirements have been identified for CPD systems, including: integration with physical testing and validation systems for “hardware-in-the-loop” simulations; and semi-automated interactive systems that involves human interventions [11]. We believe HLA to be helpful in addressing these requirements. Therefore, we will integrate these technologies in our distributed and interactive system for integrated design and simulation.

3. Integrated design and simulation in the development of complex products

3.1. Complex product development—an industrial scenario

Human element must be included in any design research methodology as new products can never be created without the actions and decisions of design engineers [1]. A design method can therefore only be successful when the design engineers are happy to accept it. In our research, the target users of our systems are those designers who are familiar with CAD/CAE/CAM tools. Moreover, the target design applications are those where simulations are necessary to achieve shorter lead time and lowered cost, i.e. the design process of complex products. Such a process generally involves the collaboration of different designers with technical knowledge from different domains, and is often influenced by downstream activities, e.g. manufacturing, service, disposal, etc. A number of characteristics of complex product development can be identified as follows:

1. The 'divide and conquer' philosophy is recommended and a multidisciplinary development team is required. Therefore, a systematic and collaborative approach needs to be developed to facilitate the division of tasks, and the integration of sub-systems.
2. Various CAE tools are used to improve the design process and simulation technology is widely applied to verify and validate design solutions at an early design stage. Therefore, integration of heterogeneous models and systems is necessary and a database is needed to store a large amount of engineering information.
3. The development is increasingly accomplished by outsourcing a significant number of components. Therefore, the collaboration with suppliers also needs to be supported and the sharing of information should be offered cautiously.

To identify the requirements for an integrated design and simulation system, we first analyze an industrial scenario of complex product development. In this scenario, the tilting mechanism of a tilting train will be developed. This mechanism consists of a hydraulic module and the tilting coach body. The ultimate goal for the design of such a system is to increase the maximum speed of a train without changing the railway infrastructure. When a train runs at a high speed over a curve, the generated centrifugal force adversely affects ride comfort. To counteract this force, we can make the coaches tilt, using a hydraulic actuator. Such a mechanism is called "active tilting" in which a control algorithm is used to achieve smooth tilting so that the degree of passenger comfort does not degrade.

Computer simulation offers great advantages in this problem, since it is very expensive and time-consuming to test design concepts by producing physical prototypes. The sketch of the tilting train and the interactions between its simulation models, namely control model, dynamics model and hydraulic model, are shown in Fig. 1. Specifically, the control model triggers the hydraulic model based on the track data and the velocity and acceleration of the coach body. The hydraulic model drives the coach body to tilt and the dynamics model calculates the velocity and acceleration based on geometric information and the torque. By simply changing the parameters of one or more models, design concepts can be tested in different operation conditions, e.g. running speeds, curvatures of the track and cants of the track, etc. Traditionally, engineers of a specific discipline create models using a single tool while neglecting or simplifying influence from other disciplines. For instance, engineers in this case study previously used simple time-varying functions to drive the coach when performing multi-body dynamics simulation, which inevitably affects the accuracy of simulation. Therefore an integrated design and simulation system is helpful in this case, which supports the integration of models and the collaboration between designers, simulation engineers and manufacturers.

3.2. Integrated and collaborative development on the WWW

Integrated and collaborative design and simulation emphasizes leveraging resources possessed by different organizations and integrating capabilities offered by different technologies. It therefore involves the coordination of people and resources and the integration of design and simulation. We envision supporting integrated design and simulation using a distributed environment on the World Wide Web (WWW), as shown in Fig. 2. The users of this collaborative development environment are the analysis

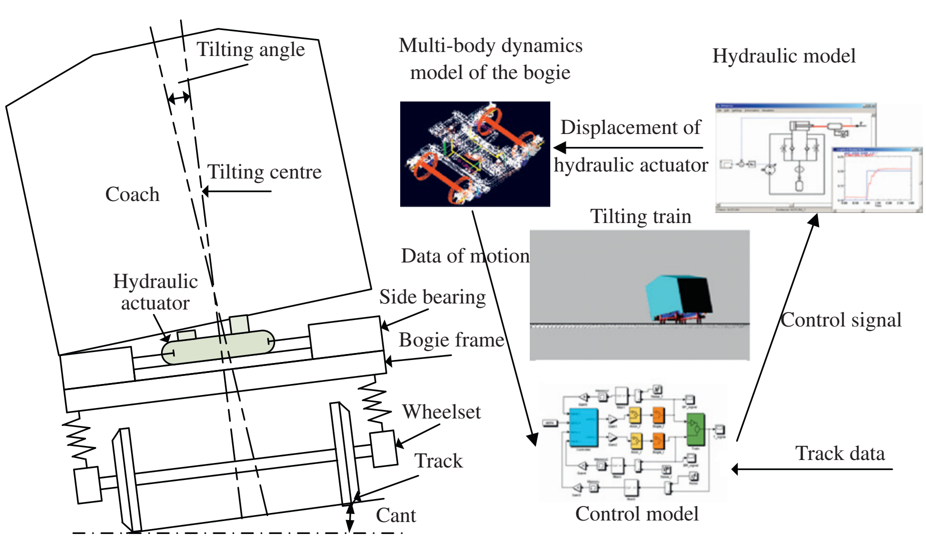

Fig. 1. Sketch of a tilting train and the interactions between its simulation models.

engineers, designers, system analysts and manufacturers, all of whom may come from departments geographically distributed. They work collaboratively in this integrated environment to create components, define layouts, identify constraints, construct simulation models and perform analysis. The development process, which can be divided as a design process and a simulation process, is performed iteratively until optimized designs are achieved based on the product requirements.

In the design process, designers create geometric models of the system by taking into account the manufacturing advices and the feedbacks from the simulation process. Form information is then provided to the simulation process, e.g. geometric information, control strategy, as well as the selection of materials. In the simulation process, form information is used to create behavior models, which are integrated in the multidisciplinary simulation, and design concepts are tested in a variety of settings. Once the simulation process is completed, useful information can be provided to the design process, e.g. requirements addressed, problems solved, new problems raised and advices learned. Apart from various tools provided by design resources and analysis resources, a supporting tool with the capability of leading users through this iterative process and supporting distributed integration and collaboration is necessary. A number of requirements are raised, including: supporting collaborative work; easy human–computer interaction; protection of model details; and integration of distributed simulation models.

4. Implementation of a distributed and interactive system

4.1. System architecture

A distributed and interactive system facilitates the development of multidisciplinary simulation which is performed by various users focusing on different tasks. Moreover, it enables the collaborative work of users by providing consistent interfaces and using unified data model. To implement these functionalities, a Web-based distributed and interactive system is designed and implemented, which leads users throughout the development process of multidisciplinary simulation. The architecture of such a system is shown in Fig. 3, which incorporates High Level Architecture (HLA) and Web Services as the infrastructural technologies. Users interact with the Web-based system at the server side by using a Web browser without the need to install software on their computers. A set of servers is utilized to provide the computational support for multidisciplinary simulation, namely Web server, model server, data server and application server. A Web server is the container where the Web-based system is deployed, and implements the data communication with end users. The Run-Time Infrastructure (RTI) of HLA and HLA-based simulation federation locate on an application server, serving the purpose of managing the simulation interactions between modules of a multidisciplinary simulation. A model server is a repository of computational resources (codes, models and tools)

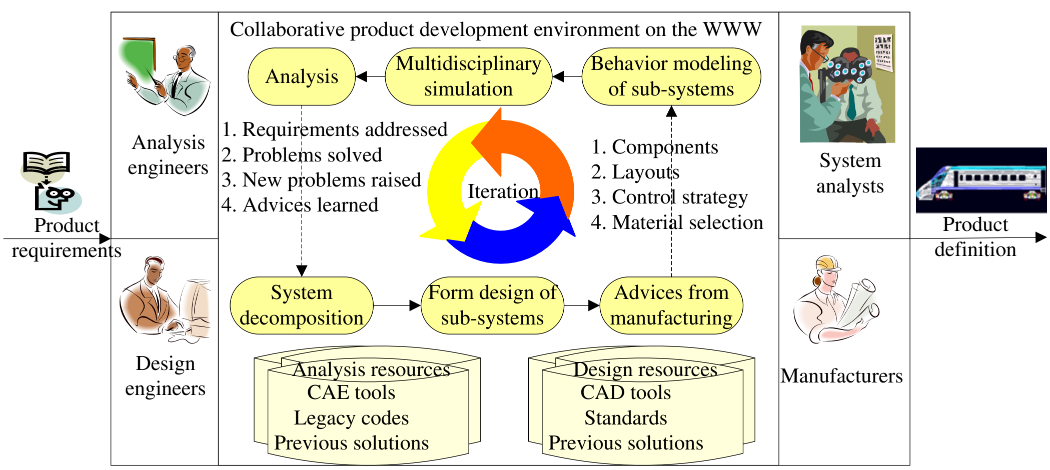

Fig. 2. Integrated design and simulation in complex product development.

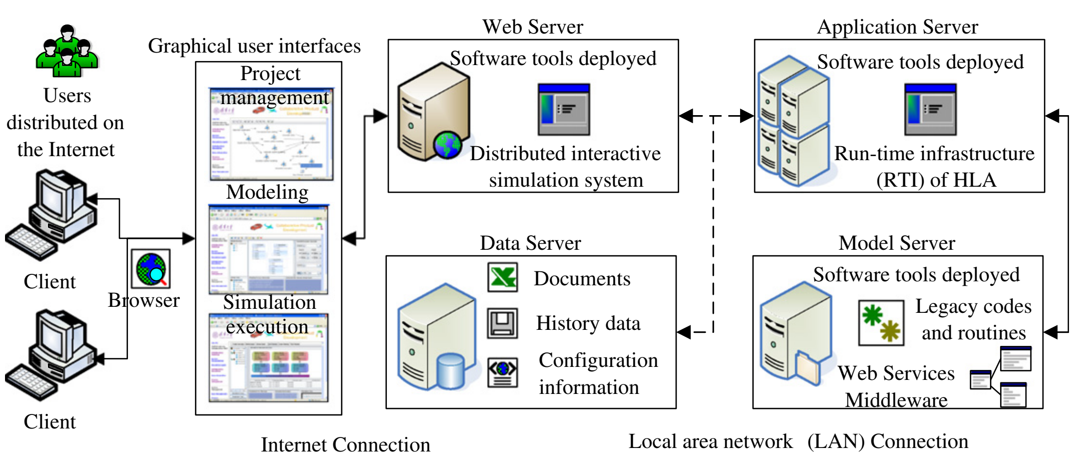

Fig. 3. Architecture of a Web-based distributed and interactive simulation system.

which are encapsulated as Web Services and accessible by the corresponding federates of a HLA-based simulation federation. A data server supplements the Web server by storing necessary data for simulation development, e.g. documents, history data and simulation configuration information.

Using the distributed and interactive system, the complex simulation problems of a physical system can be decomposed into several sub-systems which are aggregated during run-time to forecast the behavior of engineered system. The function of such a system is implemented by four separate modules as shown in Fig. 3. In this solution, key functions facilitating the simulation process are provided by the system while separate simulation models are encapsulated as Web Services distributed on the Internet. At run-time, the Web server communicates with data server and application server in the local area network (LAN) and the software components in application server communicate with the services deployed in the model server through Internet connection. Thereby, the logic of simulation interaction is separated from the execution of a single simulation model. This can achieve two-fold advantages. First, simulation interaction, which imposes significant load on network connection, is undertaken on LAN, so as to guarantee the efficiency of communication for simulation interaction. Secondly, legacy models possessed by several departments can be distributed on the Internet, which enables improved accessibility and cautious sharing of information. Central to this solution are the functions which enable the performing of simulation, and the integration of HLA and Web Services as a holistic infrastructure.

4.2. Key functions of the system to support collaborative simulation

In order to efficiently support the development of multidisciplinary simulation, the distributed and interactive simulation system has functionalities as follows:

1. A simple engine for process management, which guides the users throughout the design process. Through this engine, users can be informed about what has been done and what should be done next. Moreover, users can be assigned different roles according to the tasks they will perform in the whole process, so as to avoid unnecessary re-work.
2. A multidisciplinary modeling module should be provided to support the users in decomposing the system into sub-systems and aggregating them as a whole-system model with distributed components. Necessary information can be obtained through this module so that the deployment and execution of simulations is well supported.
3. A tool for managing the simulation execution process should be provided, which enables users to control the simulation process and perform on-line updates to simulation parameters. It consists of a manager for simulation running, a data-collector and an on-line editor.
4. A simulation post-processing module should be developed to select and store analysis data, evaluate results and highlight the issues raised during the simulation.

4.3. Run-time integration of a simulation federation based on HLA and Web Services

It is difficult to integrate simulation models that are created using different tools and distributed on the Internet during the run-time of simulation execution. First, the simulation advancement of each model may take place at any physical time while the logical time of all the models must be well-managed. Secondly, during each simulation step, accurate data-exchange must be performed to achieve accurate simulation so that the behavior of a physical system can be forecasted correctly. Thirdly, the simulation process can be started, paused and stopped in response to the command of the user, so as to provide on-line simulation control. These problems are resolved by constructing a modular infrastructure, which is based on the run-time integration of HLA and Web Services as shown in Fig. 4. On systems employing such an infrastructure, users can get involved in a simulation process by either giving control command or providing run-time data. Moreover, simulation models can be developed separately and integrated at run-time with only data transferred.

As an IEEE standard for distributed simulation, HLA provides specifications for managing distributed simulation components, and a federation integration framework for run-time interaction. In HLA-based simulation, a simulation federation is a virtual concept, which comprises several federates which are the basic units of a simulation system. The software components serving as both a federate and a Web service client are called Federate Agents (FA) in our framework. Each federate communicates with the Run-Time Infrastructure (RTI), and advances the simulation process according to its own time management strategy. RTI schedules the messages exchanged within the federation and advances the time of the whole federation only when there are no conflicts in the time-advancement requests of all federates. Apart from federates which represent simulation models and aims to accomplish data-exchange (for instance, FA A and B in Fig. 4), two more federates are added in our federation, namely; on-line controller and data collector. On-line controller, which receives commands from the user interface for simulation control and sends the corresponding requests to RTI, is a federate enabling the on-line control of a simulation process. Data collector, on the other hand, gets the data from simulation federation and presents it properly on the graphical user interface.

In this framework, HLA-based federation is used to support run-time data exchange, which runs in LAN to achieve improved efficiency and effectiveness. The simulation of each model is executed by performing numerical calculation using a suitable solver which can be provided by commercial software tools, e.g. Adams [33] or Matlab-Simulink [34]. To implement the distributed solving and interaction of these models, each model is encapsulated as Web Services which are accessed by remote service subscribers. This enables the simulation functionality (encapsulated as Web Services) and run-time interaction

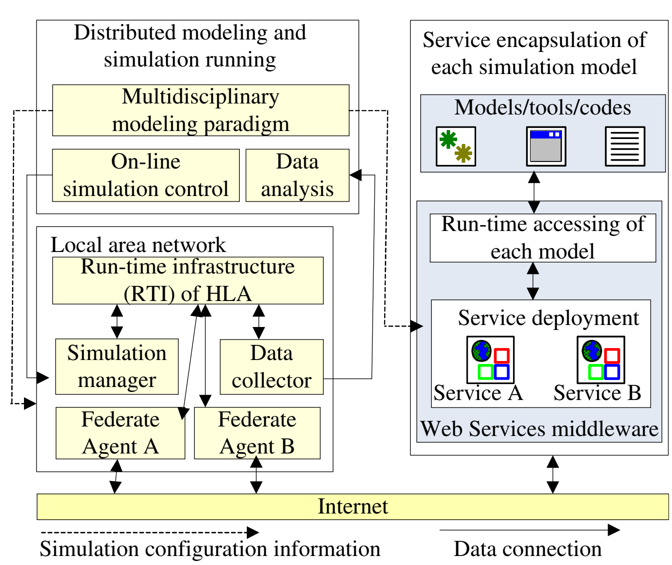

Fig. 4. Run-time integration of HLA and Web Services.

(implemented in HLA federation) to be separated as independent modules, which implements the flexible integration of a simulation system, i.e. changes will not be propagated to the internal components of each other. Thereby, the distributed and interactive features are all supported by the run-time integration of HLA and Web Services. Moreover, a multidisciplinary modeling paradigm is introduced to provide the users with a high-level tool for describing a simulation system. The information obtained in the multidisciplinary modeling is used to guide the construction of HLA federation and Web Services encapsulation and development.

4.4. Prototype implementation

A Web-based distributed and interactive system has been developed as the prototypical implementation of our solution to the integrated design and simulation for CPD. The Graphical User Interfaces (GUIs) are created using Java Server Pages (JSP) technology and Java applets. Other components of the system are all implemented using Java and the whole system is deployed via the Apache Tomcat Web Server [35]. The middleware technologies utilized to support HLA-based simulation and Web Services encapsulation are the Pitch pRTI™ product [36] and Apache Axis [37]. MySQL database management system [38] is installed in our data server to store the data generated throughout the integrated design and simulation process. Java 3D technology [39] is employed to present 3D models created using various visualization techniques. This system can be deployed on multiple operating systems due to Java's cross-platform feature. The Web-based interfaces allow end users from across the world to participate in the development without the need of installing any tools on their computers.

The system is still under further development and its current capability can provide basic support to the integrated and collaborative design and simulation envisioned by us. On the system, various users, e.g. designers, simulation engineers, etc., can work together to deliberate, create, integrate and even participate in simulations at different stages of engineering design. Specifically, four main functionalities are provided to support these tasks. First, design information, e.g. documents, visualization models, design rationale, can be licensed and shared among designers, experts and manufacturers. Secondly, users with different roles can focus only on the tasks they get involved in and work together to make decisions. Thirdly, users can control the advancement of a simulation that is implemented by integrating a number of models running on different machines. Fourthly, run-time data are recorded and a post-processing module allows users to track changes on design variables. The present system does not support the creation and deployment of Web Services that is implemented separately by software engineers. The snapshots of the GUIs for specifying interactions between sub-systems, for integrating simulation services, and for online control and monitoring of a simulation process are shown in Figs. 5 and 6 through Fig. 7.

As discussed above, integrated design and simulation is a process involving a multi-disciplinary development team consisting of designers, simulation engineers, system analysts, software engineers, manufacturers and project managers. A scenario of multidisciplinary collaborative simulation is given as follows to illustrate how various users work on the system:

1. Project managers and engineers work together to define the workflow for the development project and use it to guide team members throughout the project.
2. Design engineers, system analysts and manufacturers start to analyze the requirements of design and put specific information about basic concepts and evaluation criteria on the system.
3. Design engineers create design models using CAD tools whist getting information from simulation experts and manufacturing during this process.
4. Preliminary geometric models are uploaded to the system to be stored and visualized so that simulation engineers can work based on this information.

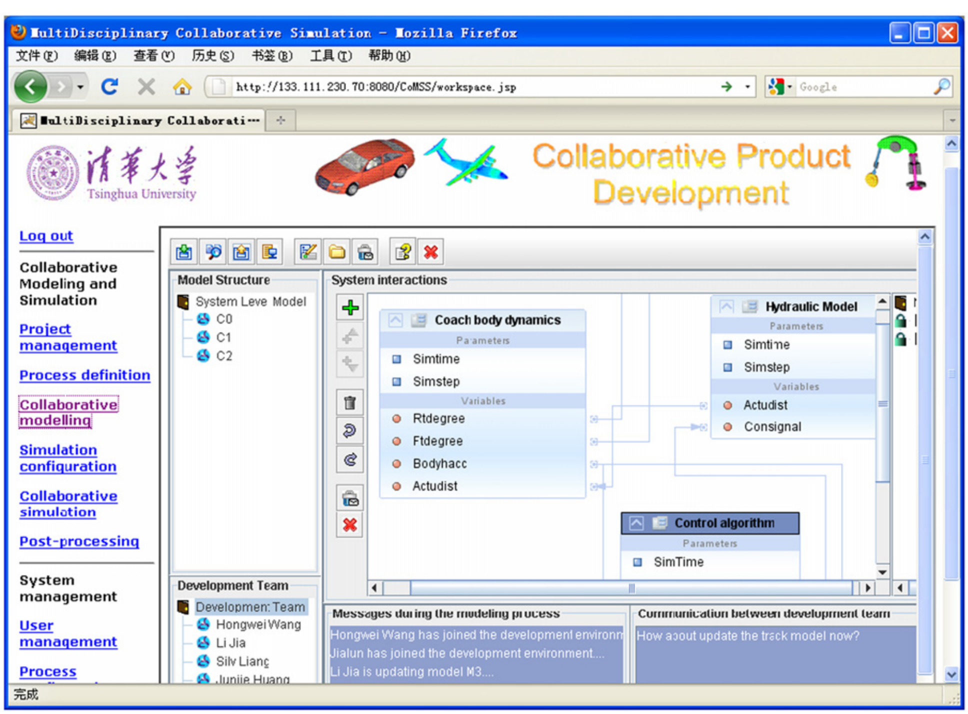

Fig. 5. Snapshot of the GUI for specifying the interactions between sub-system models.

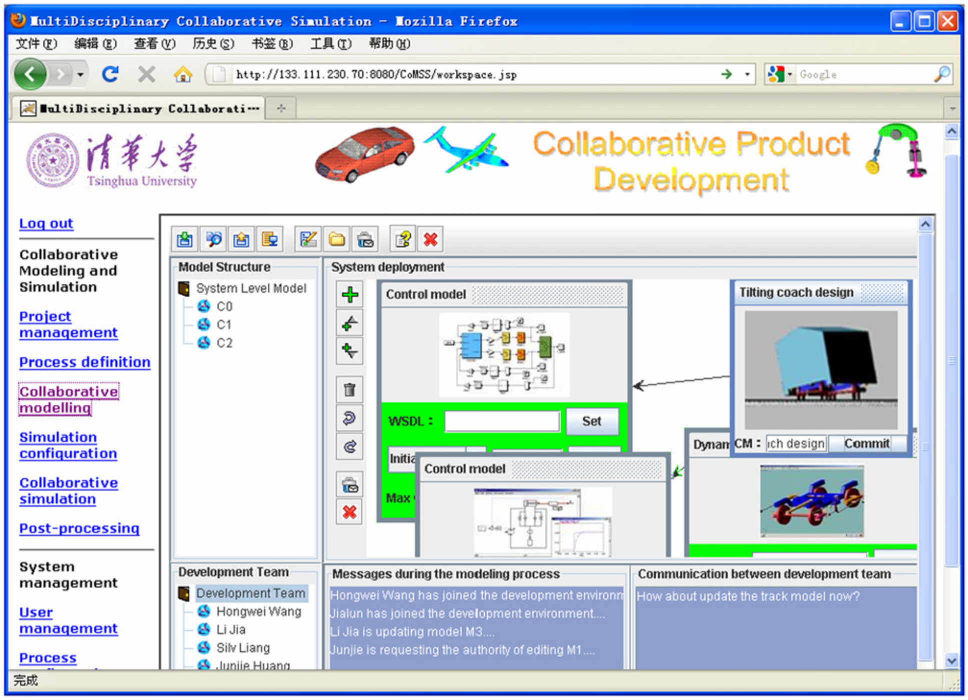

Fig. 6. Snapshot of the GUI for the integration and deployment of simulation services.

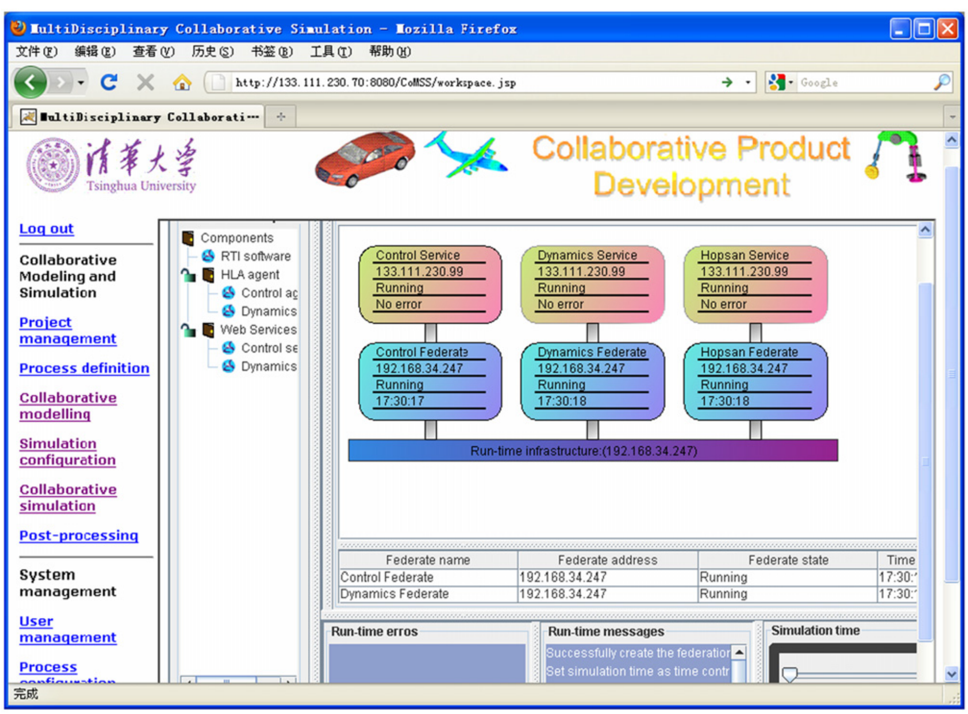

Fig. 7. Snapshot of the GUI for the on-line monitoring and control of a simulation process.

5. System analysts and simulation engineers create the high-level model of a simulation that provides information for the simulation system about how many sub-system models have been created and how they should be integrated at run-time. This task essentially involves dividing the system as several sub-systems, finding reusable legacy models, specifying the interactions between these sub-systems and solving integration issues. During this process, users can focus only on a specific part of the whole development by using a multi-view modeling framework.
6. Simulation engineers get the requirement information for the simulation of each sub-system and construct simulation models with the tool they are familiar with. Programming interfaces to the model they created should also be provided to control the simulation process of a specific model.

7. Software engineers work with simulation engineers to encapsulate the models created as Web Services and deploy the services on the Internet, and submit this information to the Web-based system.
8. System analysts and simulation engineers then work together to run the simulation via the simulation running manager interface. They can start, pause and stop the simulation just like controlling a single model. Based on the phenomena they observe during the simulation, they will provide feedbacks to design engineers and project managers via the Web-based system.

A number of methods have been developed to achieve the desired functionality so that the scenario above can be supported by the system. Specifically, a model-driven approach is proposed to bridge the high-level modeling and the infrastructural technologies. The interaction mechanism between sub-system models also needs to be elaborated to improve the performance of simulation. A uniform structured should be worked out to support the encapsulation of simulation models as Web Services. These techniques have been published elsewhere [6,7,9] and are beyond the scope of this paper. In the following section, we will present the key techniques developed for the implementation of the Web-based system, including: how federate agent can be customized; collaborative modeling supported by HLA-based simulation; and the implementation of run-time control.

5. Key techniques developed in the system implementation

5.1. Customization of federate agent

As mentioned earlier in Section 4.3, the software component called federate agent (FA) works as both a federate of a HLA-based simulation and as a client of Web Services. A HLA-based simulation with a number of interacting components is called a federation that has the capability of getting data from each component and sending them to interested components. Each component is called a federate, which publishes data they can generate and subscribe interesting data from other peers. Six kinds of management methods are implemented in the enabling tool, Run-time Infrastructure (RTI), of HLA, and the optimal use of these methods for the creation of HLA federation can significantly influence the performance. Therefore it is a difficult and complex task to develop a FA and a customization method is proposed to resolve this problem. This method essentially involves the development of a generic Java class which can be instantiated with a set of information. To identify the information needed by the customization method, we first analyze the development process of HLA-based simulation as shown in Fig. 8.

The development process of HLA-based simulation mainly involves five steps and the purposes of each step are discussed as follows:

1. Conceptual modeling. In this step, a simulation problem is analyzed and a conceptual model is generated, i.e. representing a simulation system as several components and the interactions between them. The requirements of the simulation are then transformed as constrains in the design of each sub-system.
2. High level modeling. Description of the system is given and made available to a HLA-based simulation. Simulation Object Model (SOM, capability of a federate to share data with others) and Federation Object Model (FOM, capability of sharing data in a federation) are generated.
3. Creation and transformation of domain models. Engineers with different expertise work on the creation of domain models using in-house codes or specific CAE tools. The created models are then transformed so that their simulation processes can be controlled by an external program.
4. Customization of FAs. In this step, FAs are customized by specifying a set of information and then can join a HLA-based simulation whilst controlling the simulation advancement of domain models.
5. Federation integration. The simulation running can be started once the information needed by a federation information is readily available. In federation integration, all the FAs are integrated by RTI in a distributed environment.

In the present system, six kinds of information are needed to develop a customized FA, as shown in Fig. 8. Specifically, "Data subscribed and published" are used to specify SOM for the FA. "Time management strategy" informs the FA about how it will contribute to the time advancement of a simulation federation. If the FA is time-controlled, it will advance its time following the command of RTI; if it is time-regulated, RTI shall take into account its requests when advancing the simulation time. "Starting time, ending time, time step" presents the information about time spans and time step. "Users' command" consists of the commands the FA will receive from users. "Address of the Web service and names of functions" presents information about how to communicate with a remote Web service. "Address of the Web service to

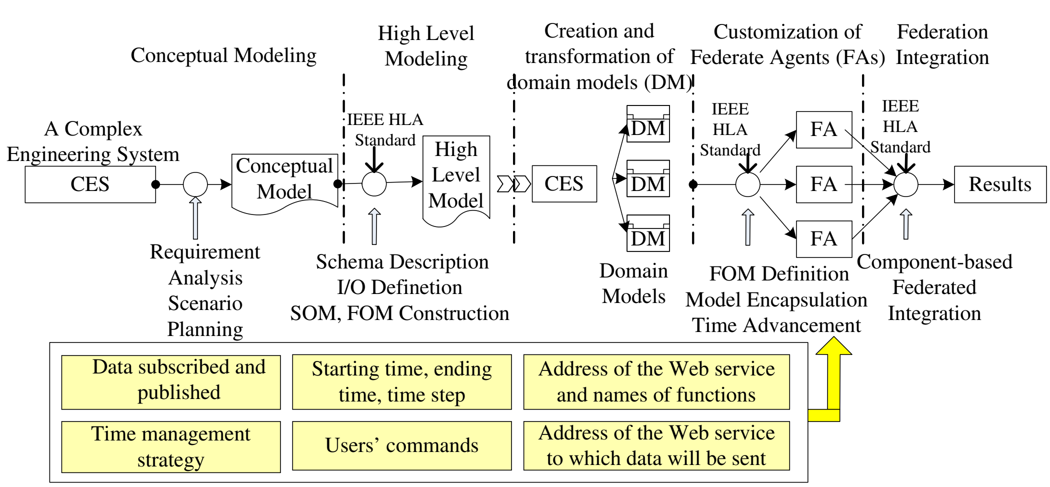

Fig. 8. A development process of HLA-based simulation.

which data will be sent" indicates to which Web Service the FA should send all the data generated during the simulation for post-processing.

5.2. Using a HLA-based simulation to enable collaborative operations on models

During collaborative modeling, multiple users need to operate on the models synchronously or asynchronously. Therefore, the modeling tool should support the collaborative operation on models in both a synchronous mode or an asynchronous mode. Moreover, users with different expertise should only be allowed to edit, or even view, a specific part of the whole model. To address these requirements, we develop a HLA-based simulation to manage the operations on models. In such a simulation federation, a FA is created for each user and will update the data in response to users' operations on models. For example, when a user wants to delete a component, he or she will do this via the GUI, e.g. by pressing the delete button. Such an action will trigger the FA to make a request to delete an object in a HLA federation that may be or not be permitted by the RTI.

The second requirement can be simply fulfilled by specifying different data subscribed and published when customizing a FA. The FA will then only receive model data the user can view, and only update model data on which the user is authorized to do so. For the first requirement, the ownership management of HLA can be used to coordinate the concurrent operations of users and avoid conflicts. In a HLA simulation, data can only be updated by one federate at one time, i.e. other federates need to wait until this federate releases the ownership. The RTI then looks up the queue and finds the next federate that will get the ownership. In our current system, a multi-view modeling paradigm is used and the ownership management is applied on each view. This means that when a user is editing one of the views of a system model, others can only look at this view while being not able to edit it.

5.3. Implementation of on-line simulation control

A simulation running is executed by advancing the simulation by a step, and finally stops when the specified period has passed. To provide more insights into the run-time behavior of a design concept, its simulation process should be controllable via an interactive on-line simulation control tool. For instance, during a simulation process, the users may want to start, suspend and stop the running to analyze data and update parameters for simulation models. A software component serving such a purpose has been developed in the Web-based system, and its structure is shown in Fig. 9. It mainly comprises of two parts, namely a simulation manager and a model operator. The simulation manager is a FA with the specific purpose of controlling the simulation process while the model operator is a service client which calls remote Web services to control the simulation of a simulation model. Such a component works in a multi-threaded way, and invokes specific threads in response to the commands received from users. For instance, if a user would like to suspend a simulation, this component will invoke the thread in which the simulation manager resides. Then simulation manager suspends the current running until further notification is given by the component. The time management strategy of the simulation manager is time-regulated so that the RTI will not grant the time advancement request of any other federates once its running is suspended.

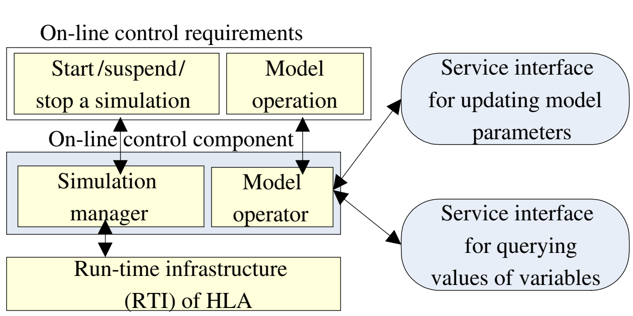

Fig. 9. Structure of a software component for the online control of a simulation process.

6. A case study

The tilting mechanism of a tilting train is analyzed previously in Section 3.1, which is developed to improve the running speed of the train. Tilting train is a typical complex product which involves several sub-systems and requires a multidisciplinary development team. As shown in Fig. 1, the development of a control algorithm depends not only on the dynamics information of the coach body but also on the track model. Using simulation technology can help test design concepts under various operating conditions before physical prototypes are produced, and achieve shorter lead time and reduced cost. The operating condition can be changed by simply changing the radii of track and modifying the geometric information of the train. The development of the control algorithm is critical to this solution as it will affect the comfort of passengers. Another two models also need to be developed, namely the dynamics model and the hydraulic model. These three simulation models will be developed by different engineers using different tools in a distributed environment, which makes it a good candidate to test our prototype system.

Six team members are invited to run this simulation on the Web-based system, including three simulation engineers, one design engineers, a system analyst and a project manager. Firstly, they work together to identify the key elements for this design as shown in Table 1, and store this information in the Web-based system. Secondly, they model the simulation system as a multi-view representation via the GUI of the system. Thirdly, design engineer create basic geometric model and put this information on the system. Fourthly, simulation engineers develop simulation models using the tools they are familiar with. In this case, the control model is created using Matlab-Simulink [34]; the dynamics model is created using Adams [33]; and the hydraulic model is created using HOPSAN [40]. After that, the system analyst helps simulation engineers to develop programs to control the simulation of each model, and deploy the simulation functionality of each model as Web services. When these Web services are available, the high-level model will be updated and

Table 1
Key variables and parameters in the tilting component development.

Name	Type	Description
a_L	Variable	Lateral acceleration of the coach body
μ	Variable	Actual displacement of the hydraulic actuator
v	Variable	Running speed of the coach body
u	Variable	Control signal to drive the hydraulic actuator
Δt	Parameter	Major simulation step of system, set as 0.01 s
t	Parameter	Time span of the simulation, set as 5 s
γ	Constraint	Tilting angle, $0^\circ \le \gamma \le 8^\circ$
$\Delta\gamma$	Constraint	Tilting rate, $0^\circ/\mathrm{s} \le \Delta\gamma \le 5^\circ/\mathrm{s}$
c	Constraint	Cant of track, $0\ \mathrm{mm} \le c \le 150\ \mathrm{mm}$
a_C	Constraint	Centrifugal acceleration, $a_C \le 0.77g$, where g is the acceleration due to gravity

the FAs will be generated. At last, all the members take part in the running of the collaborative simulation. The GUI for showing design information and simulation configuration are shown in Figs. 10 and 11.

The development process is iterated until a satisfactory solution is generated. During these iterations, different users share information and run simulation collaboratively in a distributed virtual environment. Once changes are made, the simulation can be re-started very quickly as the models are loosely coupled and the integration is well supported by HLA and Web Services. It helps achieve more effective and efficient integration by deploying simulation models as Web Services. Moreover, simulation models can be kept confidential with only the interfaces for invoking services released. The tilting angle and tilting rate of an accepted design solution is shown in Fig. 12, which fulfills the requirement highlighted in Table 1. The

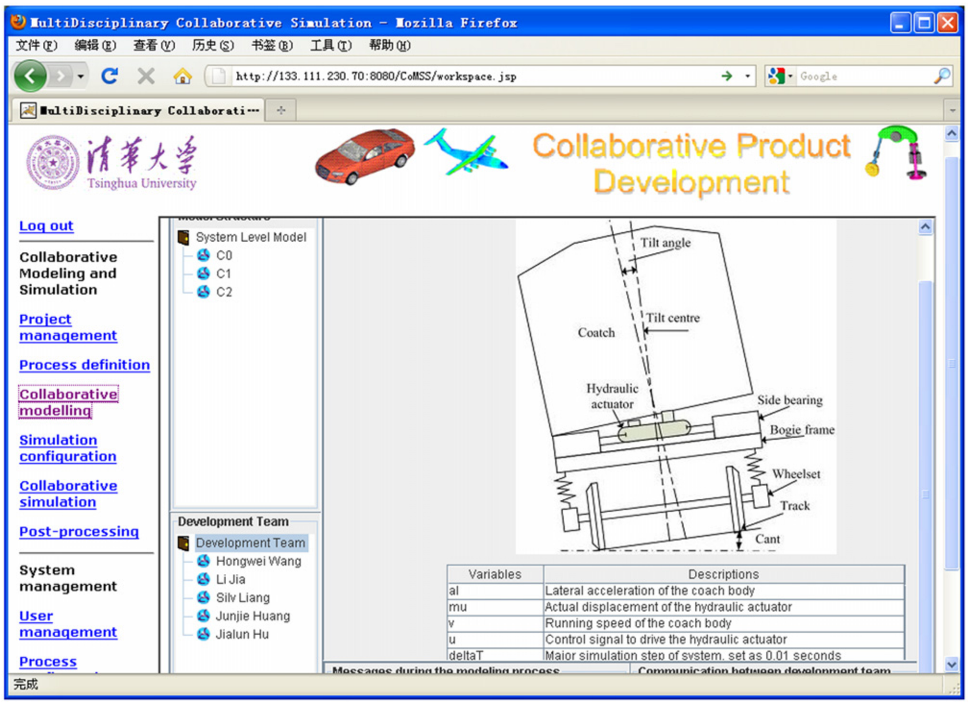

Fig. 10. Sketch and design variables shown in the Web-based GUI for the tilting train development.

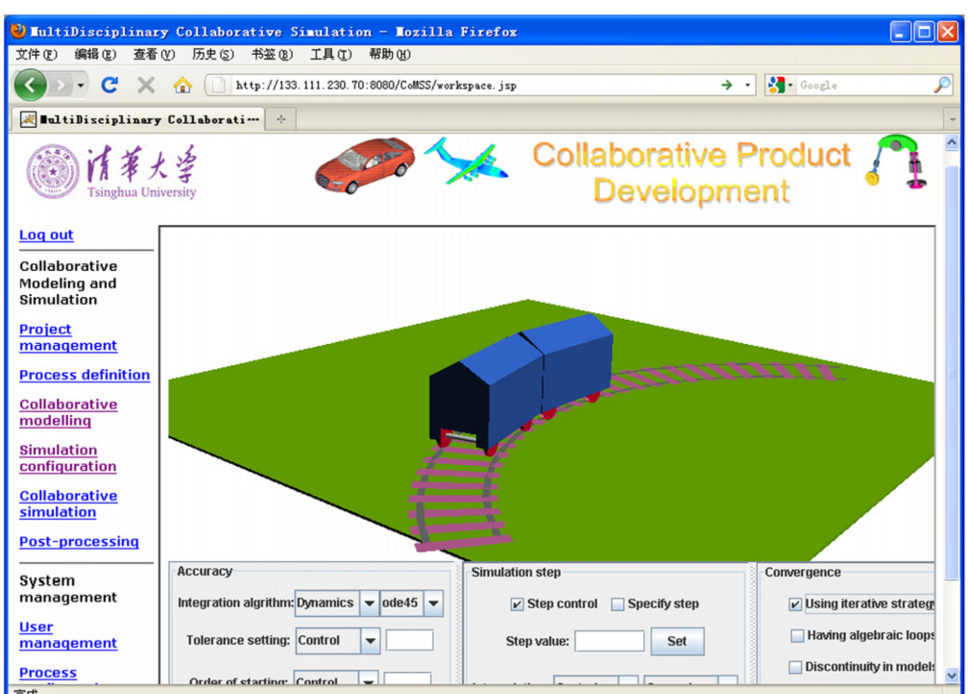

Fig. 11. Simulation configuration and model visualization in the Web-based GUI.

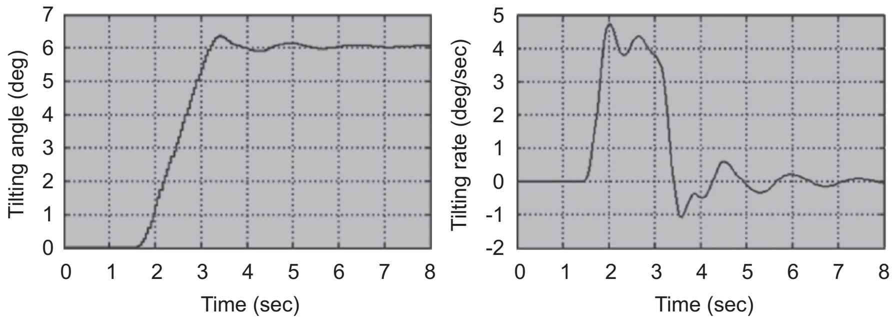

Fig. 12. Tilting angle and tilting rate of an accepted design solution.

engineers are happy with the GUIs of the Web-based system, and are impressed by the higher efficiency of development and higher accuracy of simulation compared with the traditional methods of simulation development they used.

7. Conclusion remarks

The development of modern products requires a more integrated and collaborative approach. It is already common practice to employ simulation technology in such a simulation-based development process that design and simulation are seamlessly integrated. In this context, it is necessary to develop an integrated design and simulation system for collaborative product development. Such a system will allow engineers geographically distributed to collaborate in a virtual environment. The need to develop such a system raises a number of issues, e.g. the integration of tools, the run-time interaction, the modeling paradigm, etc. All these issues need to be well resolved before this system can be applied in industry.

This paper presents the design and implementation of a distributed and interactive system supporting integrated design and simulation. An open architecture for collaborative product development is proposed and the key issues raised in the architecture are discussed in detail. A Web-based system has been developed as the prototype implementation of the proposed solution. As shown in the case study, the Web-based system can support engineers to share design information and to create, integrate, run simulations for engineering design in a distributed environment.

The major contributions of this research can be summarized as follows: 1. a framework for integrated design and simulation on the WWW; 2. a system architecture that is viable to support creating, integrating simulation models in a distributed environment; and 3. development of key techniques to implement a Web-based system, e.g. the collaborative operations on models and customized development of federate agents. The idea of using a HLA-based simulation to support synchronous and asynchronous operations on models can be also applied in other collaborative design systems. In our future work, we will continue the development work of the system to include more types of design information and run more simulation applications, and study how to improve the effectiveness and efficiency of collaborative simulation development.

Acknowledgement

This paper is supported by the National Natural Science Foundation of China (Grant no. 60674079), the Key Laboratory of Beijing Simulation Center (Grant no. B0420060524) and the National R&D High-Tech Plan (863 Program) of China.

References

[1] Wallace K, Burgess S. Methods and tools for decision making in engineering design. Design Studies 1996;16(4):429–46.
[2] Zorriassatine F, Wykes C, Parkin R, Gindy N. A survey of virtual prototyping techniques for mechanical product development. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 2003;217(4):513–30.
[3] Szykman S, Fenves SJ, Keirouz W, Shooter SB. A foundation for interoperability in next-generation product development systems. Computer-Aided Design 2001;33(7):545–59.
[4] Rodriguez K, Al-Ashaab A. Knowledge web-based system architecture for collaborative product development. Computers in Industry 2005;56(1):125–40.
[5] Shen W, et al. Systems integration and collaboration in architecture, engineering, construction, and facilities management: a review. Advanced Engineering Informatics 2010;24(2):196–207.
[6] Zhang H, Wang H, Chen D, Zacharewicz G. A model-driven approach to multidisciplinary collaborative simulation for virtual product development. Advanced Engineering Informatics 2010;24(2):167–79.
[7] Wang H, Johnson A, Zhang H, Liang S. Towards a collaborative modeling and simulation platform on the Internet. Advanced Engineering Informatics 2010;24(2):208–18.
[8] Senin N, Wallace DR, Borland N. Distributed object-based modeling in design simulation marketplace. Transaction of the ASME, Journal of Mechanical Design 2003;125(1):2–13.
[9] Wang H, Zhang H. An integrated and collaborative approach for complex product development in distributed heterogeneous environment. International Journal of Production Research 2008;46(9):2345–61.
[10] Li WD, Lu WF, Fuh JYH, Wong YS. Collaborative computer-aided design—research and development status. Computer-Aided Design 2005;37(9):931–40.
[11] Shen W, Hao Q, Li W. Computer supported collaborative design: retrospective and perspective. Computers in Industry 2008;59(9):855–62.
[12] Szykman S, Racz J, Bochenek C, Sriram RDA. Web-based system for design artifact modeling. Design Studies 2000;21(2):145–65.
[13] Kim KY, Manley DG, Yang H. Ontology-based assembly design and information sharing for collaborative product development. Computer-Aided Design 2006;38(12):1233–50.
[14] Chu C-H, Wu P-H, Hsu Y-C. Multi-agent collaborative 3D design with geometric model at different levels of detail. Robotics and Computer-Integrated Manufacturing 2009;25(2):334–47.
[15] Shen W, Hao Q, Yoon HJ, Norrie DH. Applications of agent-based systems in intelligent manufacturing: an updated review. Advanced Engineering Informatics 2006;20(4):415–31.
[16] Nahm Y-E, Ishikawa H. A hybrid multi-agent system architecture for enterprise integration using computer networks. Robotics and Computer-Integrated Manufacturing 2005;21(3):217–34.
[17] Jiao J, You X, Kumar A. An agent-based framework for collaborative negotiation in the global manufacturing supply chain network. Robotics and Computer-Integrated Manufacturing 2006;22(3):239–55.
[18] Wang JX, Tang MX, Song LN, Jiang SQ. Design and implementation of an agent-based collaborative product design system. Computers in Industry 2009;60(7):520–35.
[19] Hao Q, Shen W, Zhang Z, Park S-W, Lee J-K. Agent-based collaborative product design engineering: an industrial case study. Computers in Industry 2006;57(1):26–38.
[20] Chao K-M, Norman P, Anane R, James A. An agent-based approach to engineering design. Computers in Industry 2002;48(1):17–27.
[21] Wang YD, Shen W, Ghenniwa H. WebBlow: a Web/agent-based multidisciplinary design optimization environment. Computers in Industry 2003;52(1):17–28.

[22] Shen W, Hao Q, Wang S, Li Y, Ghenniwa H. An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. Robotics and Computer-Integrated Manufacturing 2007;23(3):315–25.
[23] Fan LQ, Kumar S, Jagdish BN, Bok SH. Development of a distributed collaborative design framework within peer-to-peer environment. Computer-Aided Design 2008;40(9):891–904.
[24] Cheng H-C, Fen C-SA. Web-based distributed problem-solving environment for engineering applications. Advances in Engineering Software 2006;37(2):112–28.
[25] Shephard MS, Beall MW, O'Bara RM, Webster BE. Toward simulation-based design. Finite Elements in Analysis and Design 2004;40(12):1575–98.
[26] Reed JA, et al. Improving the aircraft design process using Web-based modeling and simulation. ACM Transactions on Modeling and Computer Simulation 2000;10(1):58–83.
[27] Byrne J, Heavey C, Byrne PJ. A review of Web-based simulation and supporting tools. Simulation Modeling Practice and Theory 2010;18(3):253–76.
[28] Wang H, Johnson A, Zhang H. The assembly of computational models for the collaborative development of virtual prototypes. In: Proceedings of the ASME design engineering and technical conferences & Computers and information in engineering conference (IDETC&CIE 2009), San Diego, USA, 2009.
[29] Johansson B. Model management for computational system design. Ph.D. thesis. Department of Mechanical Engineering. Linköpings University; 2003.
[30] Wang H, Zhang H. Collaborative simulation environment based on HLA and Web Service. In: Proceedings of the 10th CSCWD International Conference (CSCWD 2006), Nanjing, China, 2006.
[31] Pullen JM, Brunton R, Brutzman D. Using Web Services to integrate heterogeneous simulations in a grid environment. Future Generation Computer Systems 2005;21(1):97–106.
[32] Wang H, Zhang H, Johnson A. A service-oriented approach for the collaborative simulation of complex engineering systems. In: Proceedings of the 2009 IEEE congress on services (SERVICES-I 2009), Los Angeles, USA, 2009.
[33] Adams M.S.C. 2010. Available from: 〈http://www.mscsoftware.com/Contents/Products/CAE-Tools/Adams.aspx〉.
[34] Matlab-Simulink, 2010. Available from: 〈http://www.mathworks.com/products/simulink/〉.
[35] Apache Tomcat project, 2010. Available from: 〈http://tomcat.apache.org/〉.
[36] pRTI of Pitch, 2010. Available from: 〈http://www.pitch.se/products/prti〉.
[37] Apache Axis project, 2010. Available from: 〈http://ws.apache.org/axis/〉.
[38] MySQL open source database management system, 2010. Available from: 〈http://www.mysql.com/〉.
[39] Java 3D technology, 2010. Available from: 〈http://java.sun.com/javase/technologies/desktop/java3d/〉.
[40] HOPSAN simulation tool, 2010. Available from: 〈http://www.iei.liu.se/flumes/hopsan?l=en〉.