A Virtual Experiment Platform for Mechanism Motion Cognitive Learning

Xiumin Fan, Xi Zhang, Huangchong Cheng, Yanjun Ma, and Qichang He

CIM Research Institute, Shanghai Jiaotong University, Shanghai, 200240, China
xmfan@sjtu.edu.cn

Abstract. In order to give students a more intuitionistic understanding in mechanism motion system, a virtual experiment platform is designed and developed. First, experimental component models, which contain both visual information and logical information, are built. The logical information is modeled according to a Unified Modeling Language called Modelica (MO). Then, a virtual experiment scene, which is described by Modelica, is assembled in a virtual reality environment. The virtual scene MO model is flatted to a set of equations which are compiled and solved, and the mechanism motion data can be output. Last, the motion data are exported into the Virtual Reality environment for the simulation result visualization. Students can use the platform to build mechanism system experiments and simulate the component motion for a better understanding of mechanism composition and its movement principle. The platform is universal and can be expanded to other subjects easily because the experimental components are built by Unified Modeling Method.

Keywords: virtual experiment, mechanism motion experiment, modelica modeling, virtual reality.

1 Introduction

In the traditional experiment for university teachings, a hardware experiment environment is required for students to do experiment. It is used to taking up a lot of teaching resources. In recent years, the increase of the student quantity brings a great pressure to the experiment teaching. A virtual experiment teaching platform can relieve the stress and provide approach to improve the innovation ability of students.

In the past 30 years, different kinds of object-oriented simulation tools were developed with the progress of computer technology, such as PSpice for circuit simulation, Adams for multi-body system simulation, Modelica (MO) for general multi-domain modeling and simulation. Modelica Language is proposed as an object-oriented, equation-based and unified modeling language to solve complex multi-domain physical system simulation [1][2]. In China, the research of virtual experiment for university students is in an initial stage. Some universities have obtained some results, such as, the college physics 3D online virtual experiment based on Virtools in Jilin University, the virtual experiment platform base on Jini technology in Sichuan Normal University, the remote virtual laboratory based on LabView in Taiyuan University of Technology, the virtual laboratory for intelligent

R. Shumaker (Ed.): Virtual and Mixed Reality, Part II, HCII 2011, LNCS 6774, pp. 20–29, 2011.

control based on VRML and Matlab in Zhengzhou University [3]. While in other countries, the research of virtual experiment was started earlier and many results have been achieved, such as virtual laboratory for information security teaching in Columbia State University, multimedia virtual laboratory in Swiss Federal Institute of Technology, and e-Learning virtual laboratory in Purdue University [4].

However, there is no general development platform for virtual experiment, most current virtual experiments are developed for specific subjects. This single subject development mode limits the dissemination of virtual experiment in education. In this paper, according to the consistency of the intrinsic description in mathematical equation about the physical law of different subjects [5], the object-oriented modeling language Modelica is adopted and used to build the unified models of experimental components for different subjects. Based on these experiment component models, the relationships among components are connected, and the experiment scene model is constructed, and then the experiment model is solved to get results. By this approach, a general virtual experiment platform can be constructed so as to support different virtual experiments for multi-subjects.

Based on this unified modeling and equation solving idea, a virtual experiment platform for mechanical principle course is built to support the cognitive learning of mechanism motion experiment. The organization of the following contents is: Section 2 introduces the general idea and system structure; Section 3 introduces the key methods and techniques; Section 4 introduces the development and application of the system; Section 5 summaries all the contents.

2 General Idea and System Structure for Virtual Experiment Platform

The system structure is shown in Fig.1. Firstly, basic components are abstracted from the mechanism motion components by analyzing the logical relationship among the various components of mechanism motion system experiments for mechanical principles course. The logical model of the basic component is modeled by Unified Modeling Language (Modelica). This logic model together with component's 3D model is integrated to form a completed component information model which is described by XML. The entire XML component models can be formed as a component list. Then, users can select necessary components from the component list and connect these components according to their interface types, so as to build a virtual model of the experiment and save the model into a MO file. The experimental scene files are compiled and resolved with the support of the logical MO model. Next, with the support of component's logic MO model, a solver is executed to compile and calculate the virtual experiment MO model to get calculation results. Finally, the calculation results are analyzed and useful data about the mechanical component geometric motion are extracted. These geometric motion data file is used to drive the components' movement in 3D visualization environment. The key technologies include modeling of component logical information, building and solving the experiment scene model and visualization of the simulation results. The virtual environment for virtual experiment assembly and simulation is developed based on a virtual reality application platform called "VR Flier" [6-7].

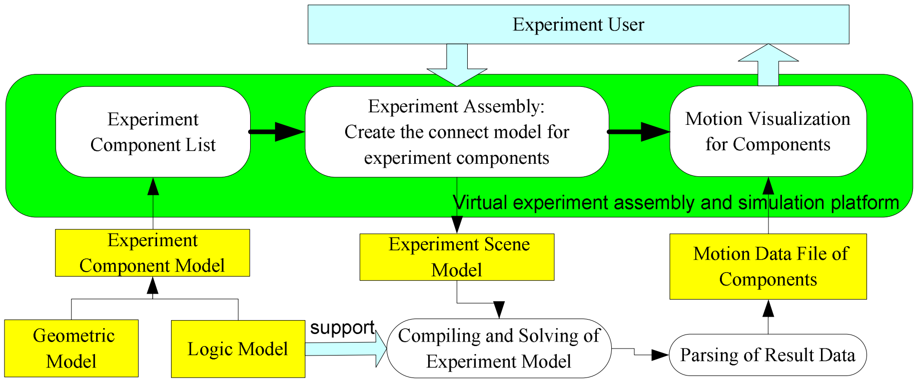

Fig. 1. System structure for virtual experiment platform

3 Method and Technology

3.1 Component Modeling and Information Mapping

Component Logic Information Modeling with Modelica. In order to build logic model of typical experiments in mechanical kinematics course, basic components that being abstracted out of these experime1wnts should be unified modeled. These basic components including bar, rotational interface, translational interface, power source, fixed translation and signal interface. These basic components are modeled in Modelica language (shown in Table 1).

Table 1. Logic models of basic components

Name of basic components	Function	Property
Bar	Define the length between two points	Two endpoints' coordinate : *x, y*; Bar length: *l*; angle between bar and x axis: *phi*
Translational interface	Constrain the relative translation motion between two components	Two endpoints' coordinate : x, y
Rotational interface	Constrain the relative rotation motion between two components	Two endpoints' coordinate :x, y
Fixed translation	Define the fixed displacement between two points	Two endpoints' coordinate :x, y
Power source	Drive a component to rotate or translate	Motion parameter output signal
Signal interface	Transfer non-motional signals between two components	Motion parameter output signal

Typical experiment components are combination of these basic components, taking the simple punching machine experiment as an example, its typical experiment components list is shown in Table 2. The numeric character for each table cell means how many numbers of basic components are included for each experiment component.

Table 2. Experiment component list for Simple Punch Machine

Basic Component / Experiment Component	Bar	translational interface	rotational interface	fixed translation	power source	signal interface
Base	0	1	2	3	0	0
Flywheel	1	0	2	0	0	1
Slider	1	1	1	0	0	0
Driver	0	0	0	0	1	1
Punch	1	1	1	0	0	0
connecting bar	1	0	2	0	0	0
Plug	3	1	2	0	0	0

Complete Component Model Description in XML Model. An experiment component's complete information, which includes component's three-dimensional visualization information and logic information from Modelica model, is described in XML (Extensible Markup Language) format. The component XML model contains five types of information as shown in follows:

(1) Component's basic information

The basic information includes the component's ID, Name, Type, Subject information, Creator, Creation date. Each component in MO library has its unique ID.

(2) Component's interface list

The interface contains following information as shown in XML format.

```
<interface>
<name></name> //Interface name
<type></type>//Type declared in Modelica
<MotionType></MotionType>//Motion style in platform (translation,
rotation or fixed)
<Position-x></Position-x>//Interface coordinate in x axis
<Position-y></Position-y>// Interface coordinate in y axis
<Position-z></Position-z>// Interface coordinate in z axis
<Normal-x></Normal-x>//Interface normal vector
<Normal-y></Normal-y>
<Normal-z></Normal-z>
</interface>
```

(3) Component's property list

The property information includes property ID, Name, Symbol, Value, Unit, Max Value, Min Value, Display Flag, Modifiable Flag and Coordinate Flag. The Display Flag defines whether the result of the property should be returned and displayed in the virtual environment. The Modifiable Flag defines whether the value of the property could be modified by user. The Coordinate Flag defines whether the parameter is related to the three-dimensional model's coordinate system.

(4) Component's logical location in Modelica library

The Modelica library contains all of components' MO logic models which are organized in a tree structure. A MO library named VElibrary has been created. A Component's MO logical location should be included in the XML description. The logical location designates MO model's full path from root node to child node, such as Modelica.VElibrary.Mechanism.MechanismAdditions.Basic.VirtualRotor.

(5) Component's geometrical information

All of Component's 3D geometric models are also stored in a model library. Component's three-dimensional models are in flt format. The same as a component's MO model, a Component's 3D model logical location should be included in the XML description, too. As the geometric modeling coordinate system may be inconsistent with the virtual environment coordinate system, a coordinate transformation matrix should also been included.

Information Mapping from Modelica Model to XML model. In the virtual experiment environment, component information models are stored in XML model format. It is necessary to map component's information from MO model to XML model. The MO model includes two parts, component declaration and function definition. As the XML model is not involved in solving process, it only needs the information of component declaration. When the solver is called and used to calculate an Experiment Scene Model, the solver will call the component's MO model which is corresponding to its XML model. The information mapping process includes five steps:

(1) Traverse the interface definition file in Modelica library to generate a **connector** class list. All of the **connector** interface components can only be called by **reference**.
(2) Traverse the interface definition file in Modelica library to generate a **partial model** class list. All the **partial model** can only be called by **extends**. The **partial model** may package several **connector** models to simplify a component's structure.
(3) A MO file may include one MO component or several MO components. Its logical hierarchy can be constructed by prefix **package**, **model**, **record**, **type**, **connector**, **block** and so on [8]. When mapping a MO file, a Model List is generated to record the components defined in the MO file. During the **Model List** generation procedure, **extends** and **reference** members in each component are traversed. **Extends** members are searched in **partial model** list and **reference** members are searched in **connector** list to get the interface list for each component.
(4) Generate a flat model of each component. A flatten algorithm [9] is adopted to get a flat model. The flat model includes parameter information and the mathematical relationship among parameters. The parameter information, including symbol, initial value, maximal value, minimal value, unit [10], modifiable flag, display flag and coordinate flag, can be extracted from the parameter declaration in flat model. For each parameter, only symbol is indispensable.
(5) After above four steps, most of the component's information in XML model is acquired. However, the parameter's name should be added by a modeler, as the name of parameter's meaning is not included in a MO model, and user can distinguish the meaning of each parameter clearly.

3.2 Virtual Experiment Scene Assembling and Calculation

The virtual experiment platform consists of two main modules: experiment assembling module and solving module.

For experiment assembling, user selects components (described as XML file) according to a certain experiment and connects the interfaces of the components (the interfaces can be connected only when the type of two interfaces is identical). For example, in mechanism kinematic motion experiments, the relative motion interface between components consists of three types: translation, rotation and logical connection; components having the same motion interface can be connected together. User can modify the parameter value of a component if its parameter is modifiable. After the experiment is assembled, user sets the simulation parameter. A MO file describing the virtual experiment scene is generated for compiling and calculation.

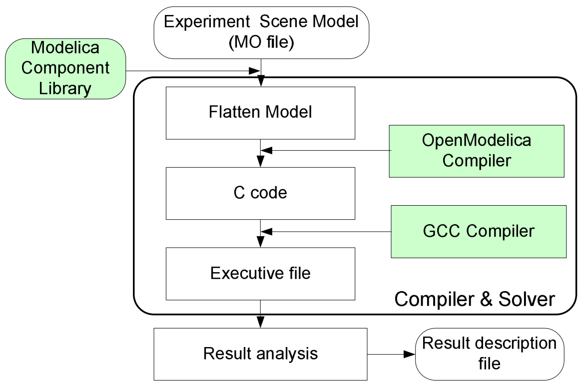

Fig. 2. Compiling and solving process for experiment scene

The compiling and solving process is as shown in Fig.2. With the experiment scene MO file ready, the Compiler & Solver load in the component's Modelica library beforehand, and then, compile and calculate the experiment scene MO file according to the simulation time setting. Final, a result description file is returned back to the virtual experiment environment. The compiling and solving process include several steps. Firstly, the experiment scene MO file is transferred to flat model. Then, the OpenModelica[11] compiler transfers the flat model into C code. Next, the GCC compiler compiles the C code into executive file. Finally, the executive file is used for result calculation.

3.3 Visualization of Calculation Results

After the experiment scene model is compiled and solved, the geometrical movement data of each moving component can be output, the data is in PLT format. In order to visualize the 3D motion of experiment components in VRFlier platform, a data conversion interface is designed for converting PLT format data to the data format specified in VR Flier, so as to drive the component's movement in virtual environment. The conversion process is as following: first, truncate the necessary data from PLT file; second, read the truncated data file into system memory through C++ I/O operation and assign the data to a corresponding array; third, transform the position coordinates and angles into transformational matrixes.

After the data processing, a path file of components' movement can be available. With this path file, a model-load-list file for loading geometrical component models can be created. The order of loading models should be corresponded to the order of the output key points. VR Flier platform can drive the movement of 3D components' models in virtual environment by loading the motion data file, and the time parameter between two animation frames can be set if necessary.

4 Development and Application of the Virtual Experiment Platform for Mechanism Motion Cognitive Learning

4.1 Development of the Virtual Experiment Platform for Mechanism Motion

Based on VR-Flier, a general platform for virtual reality application development, a virtual experiment platform for mechanism motion learning has been developed, as shown in Fig.3.

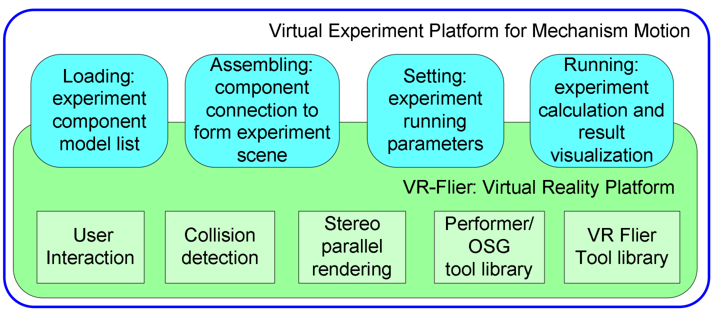

Fig. 3. Structure of software modules of the virtual experiment platform for mechanism motion

Those methods and technologies mentioned in the above section have been embedded into this VE platform. Software modules have been developed, such as "Loading: experiment component model list", "Assembling: component connection to form experiment scene", "Setting: experiment running parameters" and "Running: experiment calculation and result visualization". Some mechanism motion cognitive learning experiments, such as simple punching machine, crank & rocker mechanism and traveling guide rod mechanism, have been tested.

4.2 Assembly and Simulation of Simple Punching Machine Experiment

Taking the simple punching machine as an application example, the virtual experiment process is as follow:

Step 1: Load experiment components into virtual environment, including flywheel, base, slider, connecting bar, plug, driver and punch, as shown in Fig.4.

Step 2: Add a connection between two experiment components which have the same matching interface. For example, select the matching interface of connecting bar and plug to establish a connection, as shown in Fig.5. By this mean, all connecting components can be assembled together to build a simple puncher, as shown in Fig.6.

Step 3: Set parameter values of each component's property. Select the driver component and set its rotational speed, as shown in Fig.7.

Fig. 4. Experiment components of a simple punching machine

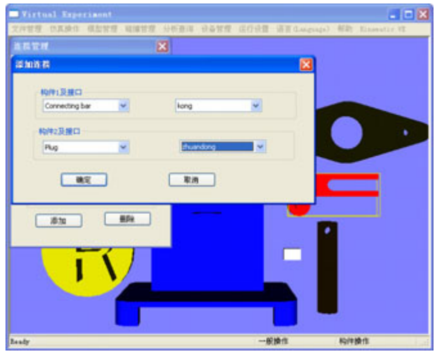

Fig. 5. Add a connection between experiment components

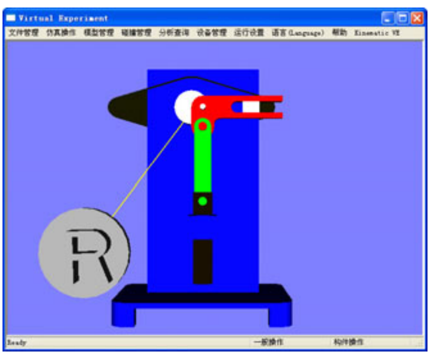

Fig. 6. All of components connected together

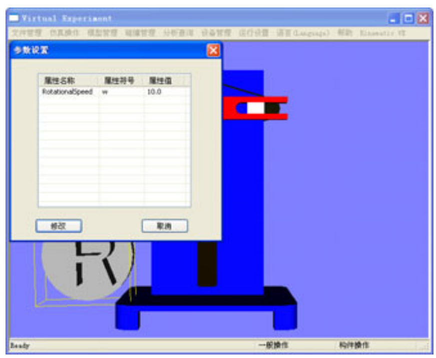

Fig. 7. Set parameters of component properties

Step 4: Set running parameters of simulation, including start time, end time and total number of intervals, as shown in Fig.8.

Step 5: Select "start" button from menu of "virtual experiment of mechanism motion" to run the experiment. The result of experiment simulation can drive the 3D component model to move and achieve the motion of punching, as shown in Fig.9.

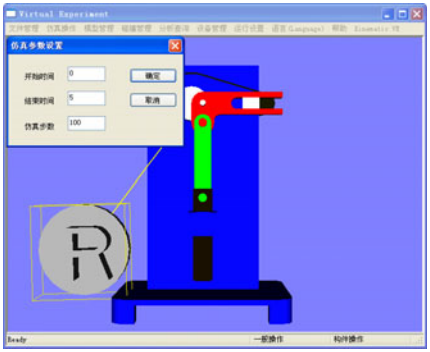

Fig. 8. Set operating parameters of simulation

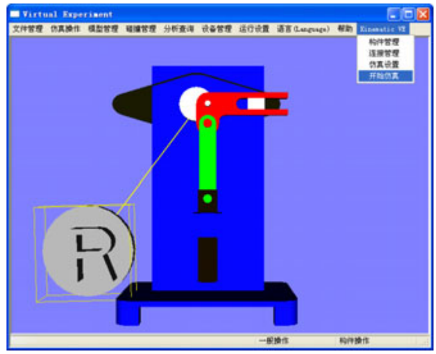

Fig. 9. Start simulation of the virtual experiment

4.3 Discussion

In this virtual environment for mechanism motion experiment, typical experiments can be assembled and simulated; it provides students with a convenient software tool for mechanism motion cognitive learning by a DIY approach. This also means the physical experiment in laboratory can be replaced by virtual experiment to a certain degree.

The development and application of the VE platform provides students with an assembly platform helpful to assemble and simulate experiment models, which enables students to better understand the assembly process of mechanism and the principle of mechanism motion. But this VE platform is still in its first version and has some shortcomings, such as the need of continuous improvement of the realism of virtual scene, and the need of a better and intuitive interaction, and even more functions, so as to let students have a more realistic feeling of doing real experimental environment.

5 Conclusion

In this paper, a virtual experiment platform for mechanism motion learning based on unified modeling method has been proposed. As the principle of mechanism motion can be described by a set of mathematical models, the experiment process can be modeled with a unified method. The method and technology, including experiment component modeling, experiment scene model assembling and solving, experiment result visualization, are discussed in detail. The virtual experiment component model contains two types of information, the geometrical information and the logical information. Basic components of mechanism motion experiment have be abstracted by analyzing the motion mechanism and modeled by Modelica language. The experiment scene MO model can be flatten to a set of equations and compiled and solved. The final experiment output data can be visualization in a 3D virtual environment. Based on these methods and technologies, together with the support of VR-Flier (a general development platform for virtual reality application), a virtual experiment platform has been developed. In this VE platform, students can assemble a mechanism motion experiment and do simulation by themselves, so as to have a better understanding of the composition of mechanism and its motion principle.

In addition, the virtual experiment platform is based on Unified Modeling Method. Three main modules, the component modeling, experiment compiling & solving, and result visualization, are independently. The structure of this VE platform is flexible and easy to be expended to experiments of other subjects, such as mechanical control.

Acknowledgments. The work is supported by the State Science and Technology Support Program of China (Grant No. 2008BAH29B03). The authors are also grateful to the editors and the anonymous reviewers for helpful comments.

References

1. Fritzson, P., Engelson, V.: Modelica- A unified object- oriented language for system modeling and simulation. In: Eric, J. (ed.) Proceedings of the 12th European Conference on Object- Oriented Programming, pp. 67–90. Springer, Brussels (1998)
2. Liu, J., Huang, Y., Chen, L., Wang, Q.: Mult -Body Dynamics Modeling Based on Modelica. China Mechanical Engineering 21(9), 1088–1089 (2010) (in Chinese)
3. Shi, L., Wang, Z., Fei, M.: Design and implementation of an intelligent control virtual laboratory based on VRML and MATLAB. Chinese Journal of Scientific Instrument, 2594–2597 (2008)
4. Averitt, S., Bugaev, M., Peeler, A., Shaffer, H., Sills, E., Stein, S., Thompson, J., Vouk, M.: Virtual Computing Laboratory(VCL). In: Proceedings of the International Conference on the Virtual Computing Initiative, May 7-8, pp. 1–16 (2007)
5. Schamai, W., Fritzson, P., Paredis, C.: Towards Unified System Modeling and Simulation with ModelicaML:Modeling of Executable Behavior Using Graphical Notations. In: Proceedings 7th Modelica Conference, Como, Italy, September 20-22 (2009)
6. Pi, X.-Z., Fan, X.-M., Yan, J.-Q.: VR Flier: A software platform for virtual reality general application development. Chinese Journal of System Simulation 17(5), 1157–1162 (2005) (in Chinese)
7. Hu, Y., Wu, D.-L., Fan, X.-M.: Virtual Reality Development Platform for Parallel Rendering. Journal of System Simulation 21(15), 4714–4721 (2009) (in Chinese)
8. Fritzson, P.: Principles of Object-oriented Modeling and Simulation with Modelica 2.1. IEEE Press, Hoboken (2004)
9. Tiller, M.M.: Introduction to Physical Modeling with Modelica. Kluwer Academic Publishers, London (2001)
10. Modelica WWW Site [EB/OL]. Modelica Group (February 10, 2006), http://www.Modelica.org
11. OpenModelica WWW Site [EB/OL]. OpenModelica (December 30, 2010), http://www.openmodelica.org/