Proceedings of the 9th International Conference on Human-Computer Interaction, Vol.1, pp. 713-717, 2001

A SYSTEM FOR SYNTHESIZING HUMAN MOTION IN VIRTUAL ENVIRONMENT

Hirotake Ishii, Kazumasa Sharyo, Daisuke Komaki and Hidekazu Yoshikawa

Graduate School of Energy Science, Kyoto University, Gokasho, Uji, Kyoto, Japan

ABSTRACT

A new designing approach based on object-oriented method is proposed to design a unique but rather complicated virtual environments the salient feature of which is that a human-shaped virtual instructor can behave like a real instructor to educate a trainee about the complicated tasks such as machine maintenance work and plant operation. The advantage of the proposed approach is that it allows the developer to construct various virtual objects for the targeted virtual environment flexibly by combining the components which are independently constructed in advance. In this paper, the elemental technologies used for the object-oriented approach are described such as to realize a virtual instructor, how to design the virtual environments with the object-oriented method and how to configure the system of executing the simulation of the virtual environment.

1. INTRODUCTION

Virtual Reality (VR) technology can be applied for various kinds of industrial and commercial fields such as amusement, education, design support and so on. Especially, various kinds of studies have been made to construct VR-based training environment for machine maintenance work and plant operation (Matsubara, 1997). The VR-based training environment has a number of advantages over the conventional training using real machines or real-size mockups. First, it is safe and economical to conduct training because virtual machines can be used instead of real machines. Second, it is possible to realize instruction functions to support the trainee to learn complicated tasks effectively. Concerning the second advantage, a new training environment has been constructed recently, in which a human-shaped virtual instructor is located to educate the trainee by indicating the trainee's error and demonstrating the complicated tasks (Rickel, 1999). The virtual instructor enables the trainee to learn complicated tasks even if the trainee is alone and a real instructor or the other trainee can not participate in the training. Therefore, such kind of training environment with virtual instructor is very promising as the next generation training environment, but there exists a problem of the workload to construct the training environment for the practical use of real field training.

Then the authors aim at developing a new construction method to reduce the workload for constructing such a next generation training environment. Concretely, the training environment is divided into several subsystems and for each subsystem a new construction method has been developed individually. In this paper, a new construction method is mainly described for human motion synthesizing subsystem which is used to synthesize the virtual instructor's body motion as 3 dimensional computer graphics.

2. ELEMENTAL TECHNOLOGIES TO REALIZE A VIRTUAL INSTRUCTOR

To realize a virtual instructor in virtual environment, 4 kinds of subsystems are necessary to be constructed by the following ways;

Subsystem1 for measuring the trainee's gesture;

To make it possible for a trainee to manipulate virtual machines, it is necessary to measure the trainee's gesture by using datagloves, polhimus sensors and so on.

Subsystem2 for deciding the virtual instructor's behavior;

It is necessary to decide the virtual instructor's behavior in according to the states of the virtual machines and the trainee's gesture. For example, if the trainee commits a wrong action, the virtual instructor should point it out and let the trainee correct the error.

Subsystem3 for the management of the virtual environment;

To make it possible for both trainee and the virtual instructor to manipulate virtual machines just like in a real world, the simulation of the interaction between virtual objects should be based on the related physical laws.

Subsystem4 for synthesizing the virtual instructor's body motion;

To make it possible for the virtual instructor to show the demonstration of the complicated tasks, it is necessary to synthesize the virtual instructor's body motion as 3 dimensional computer graphics in real time.

In the case of constructing a new training environment for different kinds of training tasks, it is necessary to reconstruct all the subsystems except for subsystem1. So a new method of constructing the virtual environment effectively is required in order to reduce the workload for constructing these subsystems. The authors aim at developing a new construction method with which a training environment for various kinds of training tasks can be constructed easily.

Concretely, for the construction of the subsystem2, the authors have developed a new modeling method for representing the virtual instructor's knowledge for machine maintenance work and plant operation by using Petri net (Endou, 2001). The representation of the virtual instructor's knowledge by using Petri net enables us to construct the virtual instructor's brain visually. For the construction of the subsystem3, the authors have developed the design support system DESCORTE to design the virtual environment in which a trainee can manipulate virtual objects just like in a real world by using human interface devices special for virtual reality (Ishii, 1999). The DESCORTE enables us to design an interactive virtual environment in which physical laws are simulated, only by the guidance of Graphical User Interface (GUI), by setting the information necessary for the training without coding programs. And for the construction of the subsystem4, the authors have developed a human motion synthesizing system AHMSS which is designed based on the idea derived from the concept of affordance, introduced by psychologist James Gibson (Ishii, 2000). The AHMSS enables us to synthesize a various kinds of human body motion easily as 3 dimensional computer graphics. These new construction methods make it possible to reduce the workload for constructing the training environment, but the following problems are remained:

1. The DESCORTE did not take into account of synthesizing the virtual instructor's motion and it is difficult to reuse the information about the virtual environment.
2. The AHMSS did not have the function of making realistic simulation based on physical laws.

Therefore in this study, the authors have developed an object-oriented method for designing virtual environment to realize the remaining three characteristics at the same time:

1. The physical laws can be simulated.
2. The virtual instructor's motion can be synthesized.
3. The information about the virtual environment can be reused to reduce the workload for constructing a new training environment.

In the section 3, the method for synthesizing human body motion is outlined, and in the succeeding sections 4 and 5, the object-oriented method and the system for simulating the virtual environment are described respectively.

3. METHOD FOR SYNTHESIZING HUMAN BODY MOTION

A human has a lot of joints and each joint has the freedom of motion from one to three degrees. So a human has a large number of posture variables. To synthesize the human motion, all of the joint's angles must be specified. Numerous algorithms for synthesizing human motions can be found in literature, but all of them are limited to use for synthesizing a particular motion. For example, the algorithm using 3 dimensional motion capture system is suitable for synthesizing the motion such as walking, gesture. But it is not suitable for synthesizing the motion to manipulate objects with hands. On the other hand, the algorithm of inverse kinematics is suitable for synthesizing the motion to manipulate objects. But this algorithm cannot be applied to the synthesis of the complex motion. Then the conventional method of developing a system using computer animation of virtual humans has been like this; first, what kinds of the virtual human's motion should be synthesized for realizing the system is decided, and then the algorithms and the data for synthesizing those kinds of virtual human's motion are constructed into the system. Although, the motion of the virtual human can be synthesized by such ways, the system is necessary to be reconstructed again when you would like to change the environment.

Then the authors have developed a human motion synthesizing system which is designed based on the idea derived from the concept of affordance. The idea is that the entire algorithms and the information necessary for synthesizing a human motion should be composed in the object database which is an archive for the virtual object's information. This design method makes it possible to use a new algorithm for synthesizing a variety of human motions without reconstructing the human motion synthesizing system. By applying the idea derived from the concept of affordance, it is possible to reduce the workload for constructing the subsystem for synthesizing the virtual instructor's body motion, but this system does not take into account of simulating the physical laws. Therefore in this study, the object-oriented method for designing the virtual environment and the simulation system for simulating the virtual environment have been developed.

4. OBJECT-ORIENTED METHOD FOR DESIGNING THE VIRTUAL ENVIRONMET

In this section, the object-oriented method for designing the virtual environment (OCTAVE) is described. With the OCTAVE method, a virtual environment is constructed by combining plural virtual objects, and each virtual object is constructed by combining plural properties of virtual objects. A virtual instructor is also constructed by combining

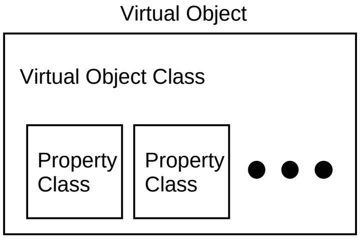

Figure 1 Configuration of a virtual object.

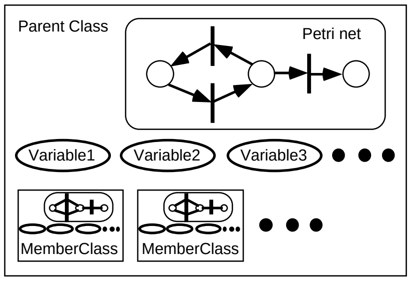

Figure 2 Configuration of a class.

plural virtual objects. Each virtual object and each property of virtual objects corresponds to “virtual object class” and “property class” respectively. Figure1 shows the configuration of a virtual object designed by the OCTAVE method. A virtual object consists of single virtual object class and plural property classes. With the OCTAVE method, as shown in Fig.2, each class consists of the following 3 elements:

1. A Petri-net for representing the discrete states of the class and their transitions,
2. Plural variables for representing the continuous state of the class, and
3. The member classes which add various properties to the parent class.

4.1 Petri-net

With the OCTAVE method, a class has two kinds of the internal state, discrete state and continuous state. The discrete state changes in accordance with the events occurred in the virtual environment. Each discrete state is defined with the process for simulating the property of virtual objects. And when the discrete state of the class changes, the process for simulating the property also changes. For example, when the state of the class for representing the property of the virtual object’s movement is “grasped by the other object which has the property ‘able to grasp an object’”, the process for simulating the virtual object’s movement is executed so that the virtual object follows the other object. And when the state of the class which represents the property of the virtual object’s movement is “still”, the process to fix the virtual object’s location and posture is executed. By this way, the transition of the virtual object’s state is simulated by changing the process for simulating the virtual object’s property.

In the case of constructing a new class, the following informations are mainly defined:

1. The kinds of states the new class can take,
2. The process executed for each state of the new class,
3. How to change the state of the new class in accordance with the occurrence of the events, and
4. The event which should occur next, after the state of the new class changes.

With the OCTAVE method, the above information can be defined by constructing Petri-net. Concretely, one place in the Petri-net corresponds to one state of the class and the existence of a token in the place represents that the class is in the state of the corresponding place. And one transition in the Petri-net corresponds to one event occurred in the virtual environment. With this method, the transition of the class’s states can be modeled as the transition of the tokens in the Petri-net.

The important point of the OCTAVE method is that the property of the virtual objects is designed not by indicating the other objects directly, but by indicating the property of the other objects. For example, to design the situation that a pen is located on a desk, the state of the pen is represented not as “located on the desk”, but as “located on the object which has the property that some objects can be located on”. By this design methodology, it becomes possible to design virtual objects independently of the other virtual objects.

4.2 Variable

With the OCTAVE method, some variables are used to represent the continuous state of the class which can’t be represented by the Petri-net alone. For example, the state that location, posture, size of virtual objects would change continuously with time.

4.3 Member class

With the OCTAVE method, a member class is used to add a property to the parent class. Figure 3 shows the relationship between a parent class and a member class. In the Figure 3, the class A, B, C has the property 1, 2, 3 respectively, and the class B and C are the member class of the class A. In this case, the class A has the properties 1, 2, 3 at the same time. By using the relationship between a parent class and a member class in this way, the parent

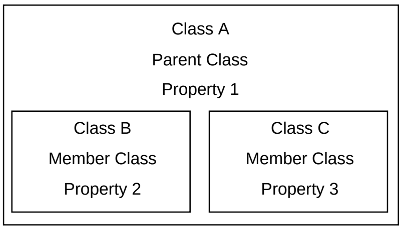

Figure 3 Relationship between a parent class and a member class.

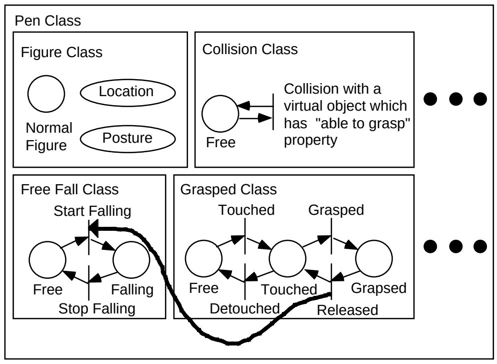

Figure 4 Example of a virtual pen designed

class can be defined by using the definition of the other classes defined in advance. This provision makes it possible to reduce the workload for designing a new class.

4.4 Simulation of interaction between virtual objects

With the conventional method for constructing virtual environments, a virtual object is indicated directly to represent the relationship between two virtual objects. For example, “a pen is located on a desk”. But with the OCTAVE method, a property of a virtual object is indicated so that it can make it possible to construct virtual objects independently of the other virtual objects. So, a new function is necessary to simulate the interaction between virtual objects designed with the OCTAVE method. In this study, a particular function is prepared to mediate the information between virtual objects. This function corresponds to “message passing” of the object-oriented concept. For example, to simulate the collision between virtual objects, a particular function is prepared to calculate the location of the virtual objects, detect the collision and inform the occurrence of the collision to the collided virtual objects. With this function, the simulation of the collision between virtual objects can be realized.

4.5 Example

Figure 4 shows an example of a virtual pen designed with the OCTAVE method. The virtual pen consists of Figure Class, Collision Class, Free Fall Class, Grasped Class and so on. The Figure Class adds the property to the Pen Class that the virtual object has shape, location and posture. The Collision Class adds the property that the virtual object collides with the other virtual object. The Free Fall Class adds the property that the virtual object falls. And the Grasped Class adds the property that the virtual object can be grasped by the other object. The variables such as Location, Posture are calculated by the simulation of the virtual environment. For example, when a token exists in the place ‘Falling’, the variable of the Location is calculated to simulate the falling state of the virtual object. The arc from the transition ‘Released’ to the transition ‘Start Falling’ means that after the transition ‘Released’ fires, the transition ‘Start Falling’ fires.

4.6 The synthesis of the virtual human’s motion

With the OCTAVE method, a virtual instructor is also constructed by combining plural virtual objects, which are constructed by combining plural classes. To make it possible to synthesize the virtual instructor’s motion, an “Affordance Class” is prepared to produce the information necessary for synthesizing the virtual instructor’s motion. Similarly as the affordance-based method described in the section 3, the Affordance Class is composed in the virtual objects located in the virtual environment. And the information for synthesizing the virtual instructor’s motion is transferred from the virtual objects to the virtual instructor by using the message passing described in the subsection 4.4. For example, in the case of constructing a virtual pen, the Affordance Class is prepared so that it would include the information for synthesizing the virtual instructor’s motion such as “grasp a pen”, “throw a pen”. And the Affordance Class is composed in the virtual pen with the other classes described in the subsection 4.5. And the virtual instructor’s motion is synthesized by using the information dynamically transfferd from the virtual pen to the virtual instructor.

5. SYSTEM CONFIGURATION OF VIRTUAL ENVIRONMENT SIMULATION

In this study, to make it possible to design the virtual environment in accordance with the OCTAVE method described in the section 4, an original script language has been developed. Figure 5 shows an example of the script language. And the simulation system to simulate the virtual environment in accordance with the script language has been developed. Figure 6 shows the configuration of the simulation system. The simulation system consists of

```
#### Free -> Grasp ####
EventNew Grasp_Event
	Variable Instance ClashHand_Instance_Var
	Trigger <(Event)HandFigure_Instance,Clash_Event> ClashHand_Instance_Var
	StateChange Free_State Grasp_State
	Function None , None
		Fire <(Event)HandFigure_Instance,ToNotClash_Event> None , None
		Fire <(Event)BallFigure_Instance,ToNotClash_Event> None , None
		Fire <(Event)HandBound_Instance,Bound_Event> ClashHand_Instance_Var , None
	End_Function
End

#### GetAffordance ####
Event GetAfordanceName_Event GetAfordanceName_Event
	StateChange Free_State Free_State
	Function None , Afordance_Name_Var
		Calc Afordance_Name_Var = <(String)Glasp>
	END_Function
End
```

Figure 5 Example of the script language.

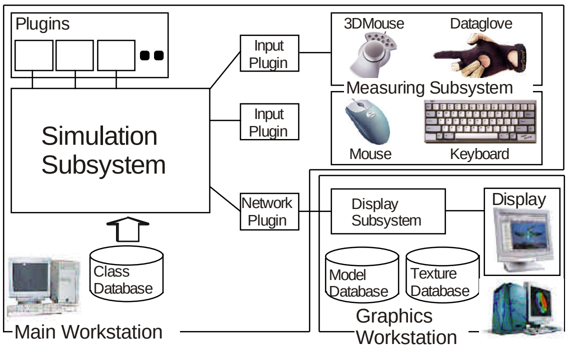

Figure 6 Configuration of the simulation system.

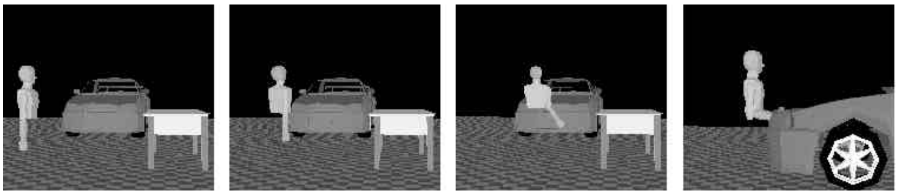

Figure 7 Example snapshots of the virtual environment designed with the OCTAVE method.

Simulation Subsystem, Measuring Subsystem, Display Subsystem and some Plugins. The Simulation Subsystem simulates the virtual environment in accordance with the script language. The Measuring Subsystem measures a trainee’s gesture and accepts the indication to the virtual instructor. The Display Subsystem generates 3 dimensional graphics of the virtual environment. And the Plugins are used to add the functions to the simulation system without reconstructing the system. Figure 7 shows the example snapshots of the virtual environment simulation designed and executed by the OCTAVE system.

6. CONCLUSIONS AND FUTURE WORKS

In this study, the object-oriented method has been proposed to reduce the workload for constructing the virtual interactive environment where a virtual instructor can behave like a real human. Based on the method, the OCTAVE simulation system has been developed to simulate the virtual environment. The proposed method makes it possible to reuse the information about the developed virtual environment, but some problems are still remained to be resolved:

1. The workload for constructing the virtual environment from the beginning is very large.
2. The high level skill is required when constructing the virtual environment according to the proposed method.

Therefore, it is considered that the system with which the virtual environment can be designed through Graphical User Interface with the OCTAVE method should be developed in the future.

REFERENCES

Matsubara, Y., Toihara, S., Tsukinari, Y. and Nagamachi, M., "Virtual Learning Environment for Discovery Learning and Its Application on Operator Training", IEICE TRANSACTIONS on Information and Systems, 1997, Vol. E80-D, No. 2, pp. 176-188.

Rickel, J. and Johnson, W., "Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control", Applied Artificial Intelligence 13, 1999, pp. 343-382.

Endou, K., Ishii, H. and Yoshikawa H., "A Study on Petri-net based Management Method between Virtual Agent and Real Human in Virtual Collaborative Environment", Human Interface Society 12th Research Meeting, 2001, (To be presented, in Japanese).

Ishii, H., Tezuka, T. and Yoshikawa H., "User-Interface Design of the Support System for Constructing Virtual Environment", Proceedings of the 8th International Conference on Human-Computer Interaction, 1999, Vol. 2, pp. 1182-1186.

Ishii, H., Ichiguchi, N., Komaki, D., Shimoda, H. and Yoshikawa H., "Development of Affordance based Human Motion Synthesizing System", Proceedings of Cognitive Systems Engineering in Process Control 2000, 2000, pp. 241-248.