

# VISUAL SCENERY TECHNOLOGY STUDY ON CYCLOTRON AND COMMUNICATION EQUIPMENT OPERATION SIMULATION

Shan Luo<sup>#</sup>, DanBo Liao, Sheng Liu, Communication Command Academy, Wuhan, China  
 ZhiGuo Yin, Fang Yang, China Institute of Atomic Energy, Beijing, China

## Abstract

Under the influence of applications using virtual reality in accelerator R&D, this paper discusses the complete process of developing Communication Equipment Operation Simulation (CEOS) based on Creator/Vega Visual Scenery Simulation Technology, operation rules modeling and solutions to its key problem. The virtual scene model for communication equipment is designed with 3D modeling software MultiGen Creator, especially its DOF technology and Switch node. The basic graphs such as wires are drawn through OpenGL call-back functions. By the virtual scene drive Vega, the application of CEOS comes true. Its simulation example shows greatest traits on building similar simulation system e.g. cyclotron virtual prototyping system as well as virtual cyclotron control system.

## INTRODUCTION

Virtual reality technology plays an important role in cyclotron development and its control system [1, 2 and 3]. Operators, as a part of cyclotron control system, their experience sometime also limiting the system performance [4]. The condition is the same in weapon equipment systems. Similar to virtual prototype system for cyclotron, computer virtual technology has advantages of reusability and flexibility over traditional one, e.g. in communication equipment training fields, this kind of training method has attracted attention from different armies since it appeared [5]. Soon it becomes research hotspot in military simulation fields. The difficulty in CEOS is similar to the one in virtual prototype system. In cyclotron virtual prototyping, a lot of the work had been done with respect to generate operation rules to make 3D device model “alive”. In CEOS, secondary development of 3D model for communication equipment is also required. Presently, related programming language/software, such as CORBA, GL Studio, VAPS and Lab View, has been widely used in virtual prototype system and equipment simulation system [5]. Models developed by these software have advantage of vivid. And independent visual scenery simulation application program can be carried out. However, the produced model data can not be directly applied to professionally visual scenery simulation software, and it is inconvenient for quadratic development [6]. The Creator/Vega visual scenery simulation technology has the merit of easy quadratic development of 3D model, which makes it possible to carry out virtual assembly and operation simulation for both accelerator and communication equipment.

## INTRODUCTION OF SIMULATION TECHNOLOGY

### Development Method of Multi-software Cooperation

In cyclotron design phase, the multi-software cooperation mode is a high-effective approach [4]. Cyclotron device e.g. the Magnet usually uses finite element software to compute 3D electromagnetic field of the structure. The outcome data provides the input for 3D mechanism design software, such as Solid Works; In the meantime, it also provides input data to beam dynamic codes, such as TRACE3D. When the latter one brings forward new requirement on electromagnetic field or structure, the input data to finite element software should be modified according to its output data and thus a new round of process of iterative appears. Both cyclotron and communication equipment have complex system structures and various equipment types, so multi-software cooperation mode is also reasonable for developing simulation system. The foundation of data exchange is OpenFlight data structure (model database in .flt file format) and ADF (Application Definition File) file.

In the described system, Creator is used to create 3D model and Creator Terrain Pro module is used to generate 3D model for virtual environment. Lynx is used to store data as Application program Define File (ADF file) needed by simulation application program. By calling Vega API, simulation program will performing pre-defined actions, corresponding to different user’s input, and following operation rules or assembly rules. Data-flow diagram for developing procedure above is shown in Fig. 1.

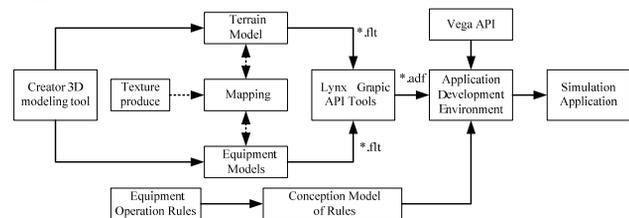


Figure 1: Data-flow diagram.

### Device Action Rules

Based on 3D device modelling, device actions should be abstracted and expressed in structured language, or Production System. The aim of the expression is to set up an action rule of 3D model of equipment entity, which enables equipment simulation program to deduce user’s

<sup>#</sup> justine1988\_li@126.com

input so as to simulate actions and reactions of equipment entity.

The first step to describe device action is to classifying it. The operation on certain component of the device may cause changes of other components' states, while its own states may also be changed. Therefore, rule expression of equipment action is composed of 5 logic parts: component, component states changes, reaction, logic association component and logic association component states changes. Equipment action can be essentially described by the set of above five components and their relations. It is quite clear and convenient to describe equipment states, and simplify and generate rule expression by using state chart (as shown in Fig. 2).

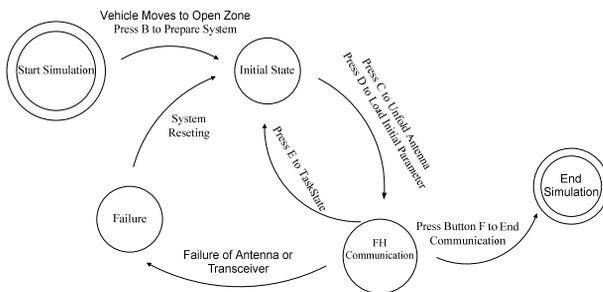


Figure 2: State chart of equipment action.

## FIELD OF APPLICATION

Develop method of multi-software cooperation and description method of device action is successfully applied to cyclotron virtual prototyping, virtual cyclotron control system development and communication equipment simulation. This paper starts with such methods being researched in cyclotron field, expands and highlights the application technology research in CEOS system, which is useful and helpful in application of cyclotron filed.

## IMPLEMENTATION AND KEY ISSUES

### Dynamic Activity of Equipment Component

DOF technology of Creator enables object to move dynamically. In DOF, a node can control its child to changes position in a given range, or rotating around a given axis with a given angle [3]. For an example, the program can set DOF for a knob on operating panel, including the axes, and rotation angle etc. parameters. This can make the knob moves in a vivid way, hence increase the feeling of reality.

### Display State Control

Switch node provided by OpenFlight model database is a special kind of node, which can control the displayed states of its children. Such nodes can include a series of mask index, which represents a set of displayed states of all children [7].

This technology can be used to change display information to represent different device states. For

example, the "power" field has 5 displayed statuses: OFF, 300W, 3kW, 7kW and 15kW. In practice, Switch node called s\_SL of 4 entity nodes can be created, and every entity node has an area to display power (as shown in Fig. 3(a)). Apply proper textual to respective areas (as shown in Fig. 3(b)). Set mask to s\_SL from 0 to 4, and at last, 5 different kinds of states can be displayed on the screen.

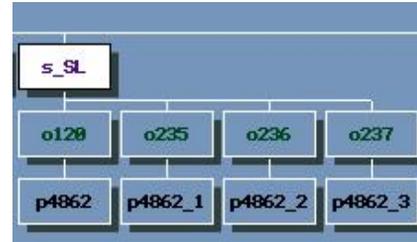


Figure 3: (a): Hierarchy structure of switch node.

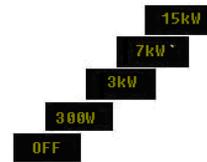


Figure 3: (b): Different states of switch node.

### Device Component Pick up

For device operation simulation, frequently it's related with a mouse click on some device parts, such as a button, a knob and so on. The corresponding term in Vega is called "pick" [8]. When user creates a pick event, the relevant function will be selected and executed by simulation program. A picker should be declared before the main Vega circle, and relevant pick mask should be set to each component of device. The main Vega circle processes all events, if triggered, the "picker" got to be executed to control DOF node and Switch node. The pseudo code for picker is shown below:

```

1. pPicker ← CreatePicker() Δcreate picker
2. if ( CheckPickEvent()=TRUE) then Δpicker
   triggered or not
   [ 1.pPickedPart ← vgGetPickerPickedPart(pPicker)
     Δget the pointer of picked part
     2.pButtonPart←vgFindPart(vgFindObj("EQUIPMENT
     "), "Button", NULL) Δget the pointer of button
     3.pfSwitch←FindSwitchInObj(vgFindObj("EQUIPME
     NT "), "Switch_Btn") Δget pointer of Switch node
     4. if (pPickedPart = pButtonPart) then
       [ 1.pfSwitchVal(pfSwitch, iSwitchMaskIndex)
         Δset mask to Switch node
       ]
     ]
   ]

```

### Special Drawings

Vega provides convenient graphics tools and abundant API to operate bottom graphics, however, it got limits. For an example, when the antenna of communication

equipment is being created, the wire is dynamically changed. It's necessary to use OpenGL in the call-back function to archive the display. The pseudo code algorithm to draw the object in Vega scene through call-back function is shown as below:

1. GL\_LINES .Color←(1.0f,0.0f,0.0f)
2. GL\_LINES .Width←(2.0f)
3. begin use GL\_LINES
  1. Locate(x1,y1,z1)
  2. Lineto (x2,y2,z2)
4. end use GL\_LINES

### A CEOS EXAMPLE

In the following CEOS implementation, terrain model and basic model have been created by Creator, e.g. vehicle workstation, radios (as shown in Fig. 4) and so on. Two different node structures, the DOF and Switch have been implemented by the Visual C++ control program, according to equipment operation rules, to fulfil the virtual operate requirement. Mean while, suitable texture and material have selected and rendered for each device component. Special dynamic objects (e.g. antenna in Fig. 5) were represent by call-back functions, utilizing Vega API.

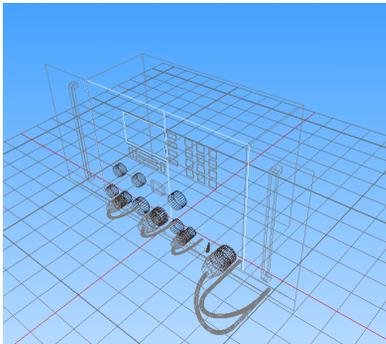


Figure 4: Basic model of communication equipment.



Figure 5: Antenna drawing.

### SUMMARY

CEOS takes advantage of multi-software cooperation development strategy, which, in turn, had been proved to be convenient and efficiently. The key technology involved can be easily carried out from engineering point

of view. In practice, the device model is vivid and has strong feeling of reality. The advantage that the program function as the actual one, can be used in equipment training as well as cyclotron control system training for operators, increasing system reliability. The example program (as shown in Fig. 6) has been tested and released to users; it turned out the response is encouraging. For further development, the program shows great potential to be applied to cyclotron prototyping system. In the same time, it can also be helpful to the virtual development of accelerator controls [3, 4].

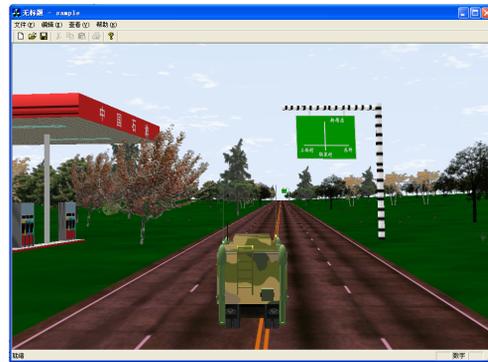


Figure 6: Simulation example of communication equipment (field).

### REFERENCES

- [1] J. F. Balaguer, de Gennaro Silvano, VENUS: A virtual reality project at CERN, Computer Graphics (ACM), 1994, 30(4): 40-43
- [2] M. Fan et al., "Applying Virtual Prototyping to the Innovative Design of Low Energy Accelerators", PAC 2003, pp. 1560-1562
- [3] Yamamoto, N. et al. "Use of a virtual accelerator for a development of an accelerator control system", PAC 1997, vol 2., pp. 2455-2457
- [4] Yin Zhiguo, etc. Virtual control system for high intensity cyclotron. Internal report of BRIF Tech. Division, CIAE.
- [5] Wang Shujie, etc. Research and Development of Virtual Simulation Interface in Windows Environment. Microcomputer Information, 2006, 22 (10-1).
- [6] Tang Jinguo, Li Zhiqin, Li Mansi. Research on the Application of GL Studio in Vega Scene.
- [7] Wang Cheng, Zhou Junqing, Li Lijun. Visual Simulation Modeling Technology of Creator. Wuhan: Huazhong Science and Technology University Publishing Company, Match, 2005.
- [8] Wang Cheng, Li Lijun, Zhou Junqing, etc. Real-Time 3D Visual Simulation Technology, Wuhan: Huazhong Science and Technology University Publishing Company, December, 2005.
- [9] Song Zhiming, Kang Fengju. The Key Technology of Visual Simulation. Computer Application, 24, 2004 (5).