

Research on V Code for Virtual Assembly System

HAN Hu

College of Mechanical Engineering
Linyi University
Linyi, China
e-mail: hanbohu@163.com

MA Huanfang

Linyi New Tian Li Machinery Co.,Ltd
Linyi, China

Abstract—Based on NC Code, in this paper, a kind of virtual descriptive language—V Code is designed and achieved for VA simulation. It is divided into three parts: scene setting code, motion controlling code and environment rendering code. At the same time, V Code definition, function classification, V Code interpreter and V Code generator are given; the grammar structure of each V Code is designed. At last, the application example of solar PECVD equipment is listed to verify V Code's effectiveness. The results show that VA simulation based on V Code is easy to understand and promote the application in many enterprises. And, it is very easy to edit equipment simulation motion program, flexible to control the simulation process, and effective to enhance the structural simulation program and improve the efficiency of the implementation process.

Keywords—V Code; Virtual simulation; VA; V Code interpreter; V Code generator

I. INTRODUCTION

In recent years, many researches on both the theories and the application of VA are conducted in many countries. Germany, United States and other developed countries lie in the leading position. Reference [1-6] mainly studied the assembly sequence planning and gave feasible algorithm. Paper [7] analyzes the human factors of the assembly process. Domestic scholars also carried out fruitful research for VA and related fields, and made many valuable new theories and new methods [8-10]. All of the literature mentioned above, which is based on secondary development for business software, mainly study the assembly sequence planning, human-computer interaction, virtual scene design and other technologies to develop VA system. Most of these studies are still in the initial stages, it is difficult for assembly workers to understand, and hard to widely use in the manufacture enterprises and really have a real benefit.

So, VA simulation based on V Code can easily set up virtual simulation environment and body structure of simulation equipment; easily edit equipment simulation motion program, very flexibly control the simulation process, effectively enhance the structural simulation program, simplify program, and improve the efficiency of the implementation process. The presentation of V Code lays a good foundation for the VA simulation in the promotion of all types of enterprises in the future.

II. DEFINITION OF V CODE

To make the virtual scene closer to the real environment, a kind of virtual descriptive language must be built to create a VA body to carry out assembly simulation, to drive parts & components of VA to assembly and disassembly, and to render virtual scene. Therefore, we have designed V Code (or Virtual languages, refers to the assembly simulation description language code), which can be divided into three parts: scene setting code, motion controlling code, and environment rendering code. Table 1 is summary table of V Code.

TABLE I. SUMMARY TABLE OF V CODE

Function	Key words including code
Scene setting	Ve01, Ve02, Ve03, Ve04, Ve05, Ve06, Ve07, Ve08
Motion controlling	Vm01, Vm02, Vm03, Vm04
Environment rendering	Vs01, Vs02, Vs03, Vs04, Vs05, Vs06, Vs07, Vs07, Vs08, Vs09, Vs10

Motion controlling code is a VA code in a virtual environment to realize parts & components' assembly process, which mainly include the following assembly movement types: linear motion with a given speed, circular motion, rotation motion, spiral feed motion around a given axis, and any path of movement.

e.g.

Vm02 (para 1, struct 1, struct 2, para 8, para 9): circular motion

para 1: No. of part, assign as a string;

struct 1: para 2-4, ordinate points on the axis, assigned to a structure (or class), which contains three floats;

struct 2: para 5-7, vector, assigned to a structure (or class), which contains three floats;

para 8: velocity (Can be regulated with a positive or a negative), assigned to a float (velocity should not be equal to zero)

para 9: angle, assigned to a float (angle should be greater than zero)

III. V CODE INTERPRETER

V Code has two text files: equipment structure description file which suffix is .vec, is made up of Ve Code, and equipment motion description file which suffix is .vmc, is made up of Vm and Vs Code. A .vec file may correspond to multiple .vmc files, because the composition

structure of one VA may be applied to a variety of assembly programs.

According to the design of V Code and the condition to be given, the language symbols which are easy for users to understand are designed, and one keyword is set for each function code. V Code interpreter is in fact to identify and deal with these keywords and at the same time extract its information. Within the interpreter program, these keywords don't appear in terms of string, but assigned to numeric code. In addition to these command keywords, two other special characters also need to specify the internal digital code, one is end of line EOL (End of line), the other is end of the process character FINISHED. Their internal numeric code is shown below.

```
#define Ve01      1
. . . . .
#define Vm01     11
. . . . .
#define Vs01     21
. . . . .
#define EOL      41
#define FINISHED 42
```

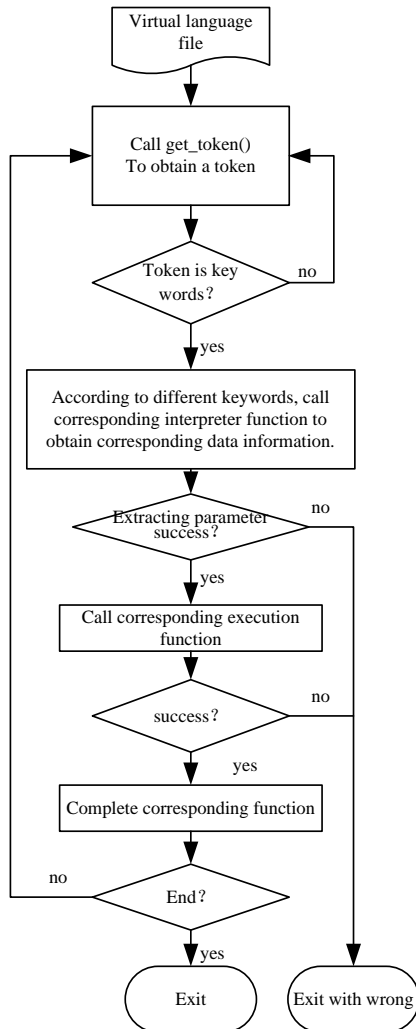


Figure 1. Flowchart of V Code interpreter

In order to easily transform the external representation (string) of a keyword into internal representation (number), these two representations should be put together

comparatively, and be saved in an array of structures named table, which is shown as follows.

```
struct commands
{
    char command[10];
    char tok;
}table[]={
    "ve01",Ve01,
    "ve02",Ve02,
    .....
};
```

All of the interpreter program is achieved through the highest level loop within the main function. Each loop is to extract a keyword from the next V Code. If the instruction is a single one, we can directly call the corresponding function to deal with; if some of the functions are completed through two or more corresponding instructions being interpreted in common, we need to judge in the main loop. In the process of reading the language files, once keywords is faced, program will execute their own function, assign the corresponding information to the corresponding data structure, and call the appropriate function to achieve their own function (interpreter flowchart is shown in Fig .1).

IV. V CODE GENERATOR

A. Automatically generate from CAD software

Shown as Fig .2, the main interface system prompts the user to press the "output *wrl* model and scene setting code" button, the system will call the function of UG secondary development to automatically generate a given precision *wrl* model, the structure file of equipment and the virtual scene setting code file which need to call to comprise the equipment.

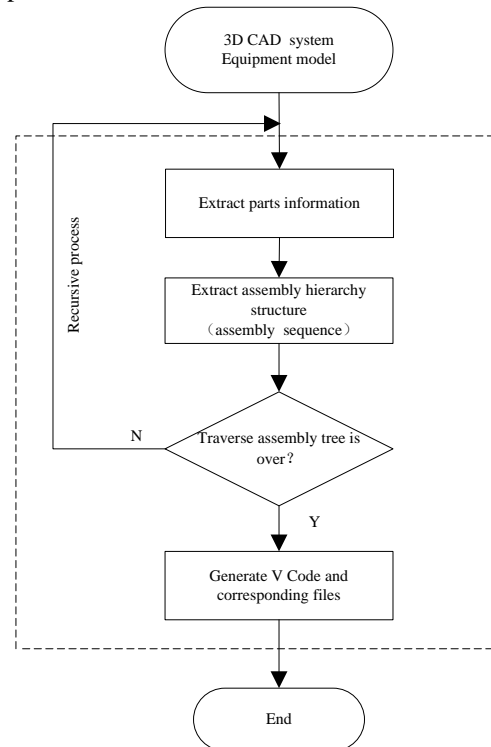


Figure 2. Flowchart of automatically generate V Code from CAD

B. Manual preparing way

According to defined code format, manual preparing is carried out in such text editor software as Notepad. Parts & components code files and motion code files which are transferred are respectively saved as .vec and .vmc files. (Writers should have experience with the assembly process.)

C. Generate through dialogue box

This way mainly is used for linear motion, circular motion, rotation motion and spiral feed motion and other motion codes, the system will automatically generate their own format. So long as we add some motion parameters, the system will automatically give the coordinates of the center of rotation (The initial default for the parts center position), which can also be modified. The workflow of the VA in this way is shown as Fig .3.

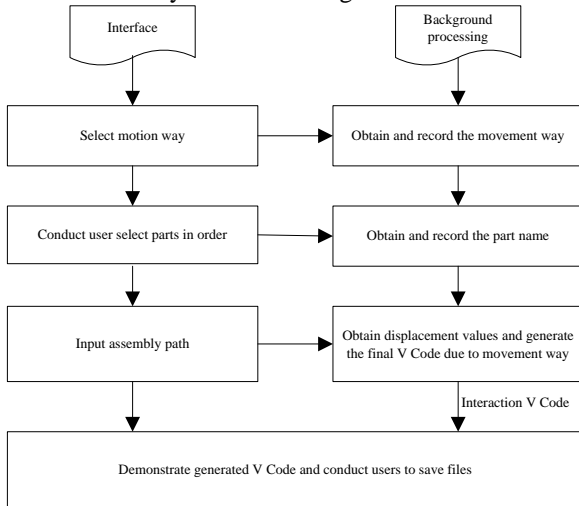


Figure 3. Interface and background processing way of interaction VA

D. Generate V Code through selecting objects and drag them by mouse

By means of the way of 3D interaction, select the equipment body needed to move by mouse, specify motion way or the way of random path, drag it to the target location, the system will automatically generate motion code, or directly move the object to the target location through the data glove, the system will automatically generate motion code, too.

V. REALIZATION AND APPLICATION OF V CODE FOR VM SIMULATION

After executive function obtains the parameters transferred by V function, the system will transform these parameters into the scene settings and the process animation. According to initial conditions and computer graphics, we can obtain the transformation matrix of linear, rotary, spiral feed and other motions; from these transformation matrixes, we can come to the scene information at any time and achieve animation process by continuously updating the scene. Add and modify the scene information by virtual statement, and complete the whole animation process model through time scene model. After executive program will traverse each time node of time chain, read the node information, clear the scene

information on the front node before updating the scene. In order to achieve smooth movement, according to the human eye's response interval, set the scene updated 25 times per second, the virtual control function is responsible for interpolating continuous motion into the discrete and exact value at time point. Fig .4 shows the implementation process of the V Code.

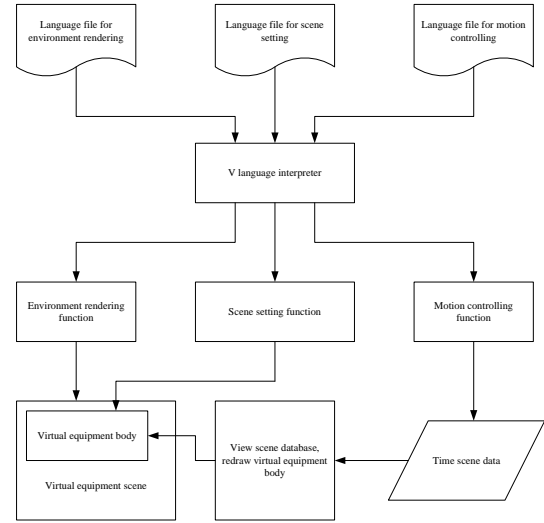


Figure 4. Flowchart of V Code realization

Keywords of virtual control functions respectively correspond with the keywords of V Code, including scene setting function, process simulation executive function, auxiliary function, v function transforms the process planning information which the V Code described, into the position & attitude, and other information corresponding to display which parts & components of VA lie at each time point. On this basis, the system calls the corresponding executive function of process simulation scene module to execute real-time rendering scenes and to achieve equipment and process simulation.

VI. APPLICATION EXAMPLE

The model library was set up using VRML in this simulation environment. It is visual through reading the library. In essence, this kind of assembling has come true the mapping of two levels, namely bottom level maps products physical model by digitized model and top level maps the true assembling process with VA course. The bottom level can implement the project analysis and assemble simulation. The top level can be carried on the products' assembly planning, verification and appraisal. Virtual assemble will realize two goals.

a. To prove the designing result and offer reference information.

b. Assembling planning, which accords with the real assembly craft most to produce, is used for guiding production.

PECVD equipment is one of the key equipment of semiconductor manufacturing process, which commonly used to manufacture silicon nitride (SixNy) or silica (SixOy) thin films in microelectronics, optoelectronics, micro-mechanical devices. According to solar battery's high-volume, low-cost production requirement, the equipment system uses linear microwave plasma PECVD deposition technique, uses SiH4 and NH3 plasma to

generate under microwave electric field and to improve reaction deposition efficiency and film performance. At the same time, the equipment makes use of continuous operation of a pipeline mode to improve productivity per unit time. PECVD equipment is made of input film flat, input film room, craft room, output film room, output film flat, and other components. 3D model of the system parts & components is built based on the above structure. In this system, we use UG NX software to build 3D model of parts & components of the PECVD equipment, and then proceed to the assembly, the assembly model is shown in Fig .5.

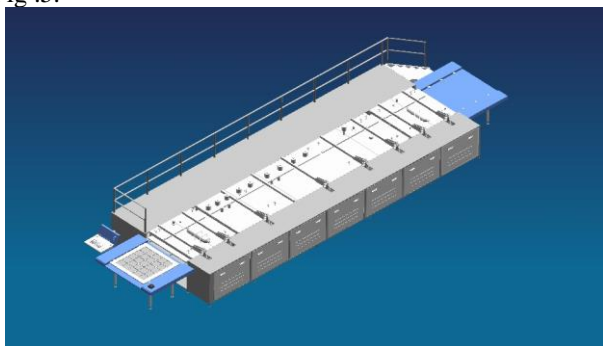


Figure 5. solid model of PECVD

After completion of the PECVD models, we use VRML file interface program for the conversion. By V Code interpreter, update Scene_of_Time, the visual simulation assembly process of PECVD equipment can be achieved. Explosion in the assembly process simulation scene is shown in Fig .6.



Figure 6. Explosion of PECVD assembly process

VII. CONCLUSION

This paper designed and implemented a kind of virtual descriptive language — V Code, for VA simulation. Focusing on V Code, we mainly do the following works.

(1) Given V Code definition and function classification, designed grammar structure of each V Code. V Code, a kind of virtual descriptive language, which can create a VA body to carry out assembly simulation, can drive parts & components of virtual equipment to assembly and disassembly, and can render virtual scene. According to the function, we divide V Code into three types: scene setting code, motion controlling code and environment rendering code.

(2) Designed V Code interpreter. According to the design of V Code and the condition to be given, the language symbols which are easy for users to understand are designed, and one keyword is set for each function code. V Code interpreter is in fact to identify and deal with these keywords and at the same time extract its information. Specially designed "Update - time" data structure to save the state (position, attitude and display attributes) changes of the parts & components at each time point.

(3) Developed V Code generator. V Code generation ways are: automatically generate from CAD software; manual program; dialogue program by dialogue box; generate motion code through selecting objects and dragging it by mouse. We can randomly select any way due to actual situation.

(4) Analyzed the VA simulation process based on V Code, finally taken solar PECVD equipment as the application example to verify assembly simulation. The results show that the VA simulation based on V Code is easy to understand and promote the application in many enterprises.

ACKNOWLEDGMENT

The project is supported by the Science and Technology Program of Shandong Provincial Education Department, (Grant No. J12LB69), partially supported by the Project of Linyi 2014 Science and Technology Program (Grant No. 201414028).

REFERENCES

- [1] Bullinger H J, Richter M, Scidel K A. VA planning. Humman factors And Ergonomics In Manufacturing, 2003, 10(3), pp.331 – 341.
- [2] Siddique Z, Rosen D W. A virtual prototyping approach to product disassembly reasoning. Computer-Aided Design, 1997, 29(12), pp.847 – 860.
- [3] Yuan Xiaobu. Interactive assemble planning in virtual environments. Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000, pp. 1462 – 1467.
- [4] Kuehne R P, Oliver J H. Virtual environment for interactive assembly planning and evaluation. Proceedings of ASME Design Engineering Technical Conference. New York, USA, 1995, pp.863–867.
- [5] Heger R, Richter M. Advanced assembly planning using virtual reality techniques, Proceedings of European Conference on Integration in Manufacturing 97, Dresden, Germany, Tech Univ Dresden, 1997, pp. 73–82.
- [6] <http://www.mel.nist.gov/>. National Institute of Standards and Technology.
- [7] Allen C, Karam K Z, Le Cam P et al. Application of Virtual reality devices to the quantitative assessment of manual assemble forces in a factory environment Proceeding of the 1995 IEEE 21th international Conference on Industrial Electronics, Control, and Instrumentation(IECON), 1995,pp. 1048 – 1053.
- [8] LI Yuan, ZHANG Tao, YU Jian-feng, et el. Research on Assembly Simulation Based on Operation Models. MECHANICAL SCIENCE AND TECHNOLOGY. 2000, 19 (3), pp. 503-505.
- [9] WAN Hua-gen, GAO Shu-ming, PENG Qun-sheng. VDVAS: An Integrated Virtual Design and VA Environment. Journal of Image and Graphics, 2002, 7(1), pp. 27-35.
- [10] ZHANG Lin-xuan, TONG Bing-shu, XIAO Tian-yuan. Realization and Application of Virtual Simulation System in CE environment. Computer Applications, 2000, (2): 194-197.