

# CROSS-PLATFORM ADAPTIVE GIS SYMBOLIZATION RESEARCH

SU Kehua<sup>a</sup>, ZHU Xinyan<sup>a</sup>, KONG Fanmin<sup>b</sup>

<sup>a</sup>State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University - skhemail@163.com, geozxy@126.net

<sup>b</sup>Scomputer and information college, Yang-En University, FuJian -

Commission VI, WG II/6

**KEY WORDS:** Symbol, Symbol Virtual Machine, Virtual Machine Plug-in, Symbol Design Tools, Symbol Sharing

## ABSTRACT:

This paper proposed a new method for sharing symbols between different GIS platform by using virtual machine technology, the symbol system consist of three parts, they are symbol virtual machine, virtual machine articulated plug-in and symbol design tools. Symbol virtual machine is a software simulated machine, it has virtual hardware, such as processor, stacks, registers, and the corresponding instruction system, symbols are described by these instructions. symbol virtual machine is independent of any specific GIS platform, responsible for dealing with GIS platform-independent function of symbolize; symbol virtual machine articulated plug-in connect the symbol virtual machine with the GIS platform, so that the GIS platform can call symbol virtual machine to do symbolize work; symbol design tools used in the production of symbols, when produce symbols we only target at symbol virtual machine, not target at specific GIS platform, thus the symbols can be shared in all the GIS platform articulated symbol virtual machine.

## 1. INTRODUCE

Cartographers and geographers use symbols on maps to represent various geographic phenomena involving location, distance, volume, movement, function, process, correlation, etc<sup>[1]</sup>. These phenomena can be classified into four basic categories: point (non-dimensional data), line (one-dimensional data), area (two-dimensional data), and volume (three-dimensional data). Once geographic features and data have been selected, generalized and classified for the map, it is necessary to choose the appropriate graphic representation or symbols for the information. Symbols have characteristics that can be manipulated to suit the category of data being mapped. These characteristics are referred to as visual variables or visual resources. Visual variables include symbol size, shape, orientation, pattern (texture), hue (colour), and colour value (brightness and lightness)<sup>[2]</sup>. The process convert the spatial data to map symbols and achieve visualization is called symbolization. Map symbol plays a key role in the symbolization, it is an important component of GIS. At present, all GIS software in the market have achieved their own map symbols system respectively, but these are only aim at specific platform, do not have interoperability, often the symbols designed for a GIS platform can not be used in another GIS platform. In addition, sharing and integration of multi-source GIS spatial data has become a goal, the majority GIS platform has provided the public exchange form for the sharing space data, but in the process of spatial information conversion inevitably lost some information, in which loses fiercely is the map symbol representation information<sup>[5-12]</sup>. For the above two reasons, exactly the same spatial data in different GIS platforms are often unable to obtain consistent representation, with the GIS development and standardization, this will increasingly become a serious problem. So we need a valid method to share map symbol libraries in different GIS platform. this paper proposed a new method to share map symbol libraries in different GIS platform by using symbol virtual machine.

## 2. BASIC PRINCIPLE OF CROSS-PLATFORM SYMBOL

In computer science, a virtual machine (VM) is a software implementation of a machine (computer) that executes programs like a real machine<sup>[3]</sup>. Virtual machines are separated in two major categories, based on their use and degree of correspondence to any real machine. A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS). In contrast, a process virtual machine is designed to run a single program, which means that it supports a single process. Virtual machine addresses this challenge by enabling the creation of platform-independent programs. A single virtual machine based program can run unchanged on a wide range of computers and devices. Compared with programs compiled for a specific hardware and operating system, platform-independent programs written in for virtual machine can be easier and cheaper to develop, administer, and maintain<sup>[4]</sup>. By using virtual machine technology we can share symbols between different GIS. The basic principle is: When we create symbol libraries, we do not create symbols for the special GIS platform but for the symbol virtual machine. Because the symbol libraries does not create for special GIS platform, if we want use this kinds of symbols in a special GIS platform, we must implement a symbol virtual machine for that special GIS platform, once the symbol virtual machine has been implemented, all the symbols designed by this method can be used in that special GIS platform. In a short word, we share map symbol libraries in different GIS platform by symbol virtual machine. Using the symbol virtual machine, we screened the difference of the concrete GIS software platform, and provide a uniform symbol interface for the GIS symbol designers. In order convenient to implement, we can define a public symbol virtual machine, and then for different GIS platforms implement a symbol plug-in, which connect the symbol with the special GIS platform. The entire symbol

system consist in three part: symbol virtual machine, symbol plug-in and symbols design tools.

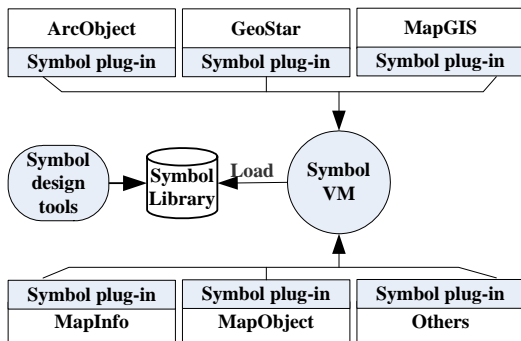


Figure 1. Component of virtual machine based symbol system

### 3. SYMBOL MACHINE

The virtual machine design process itself is a symbol, symbols virtual machine implementation of such procedures is the machine, but in reality, this machine is not actually exist, it is achieved through software simulation, it is unimaginable in the hardware, Such as the processor, stack, such as registers, but also with the corresponding instruction. Symbol virtual machine can not leave the host GIS environment and the implementation of a separate operation, it must be called to get the host GIS implementation, it is the host GIS there are two aspects of the Internet, on the one hand the need to host GIS spatial data transmission symbol Virtual Machine, On the other hand symbol virtual machine need to sign the results back to the host GIS, this information is through the symbol of interactive data interface and graphics rendering interface (see Figure 2), when the space to host GIS data symbols And, it will notify the symbol virtual machine to load the symbol, and symbols of the interface through the spatial data transmission symbol virtual machine, the virtual machine from the implementation of the implementation of the engine will be loaded into memory in order for the symbol of symbols, In the implementation of the encounter in the course of mapping operation, the implementation of graphics rendering engine will be passed to the callback interface software for GIS mapping function, thereby rendering to the graphics software for GIS mapping devices.

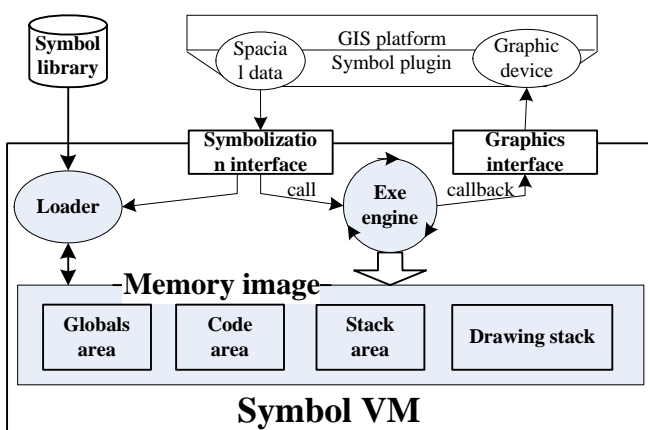


Figure 2. Architecture of symbol virtual machine

#### 3.1 Memory image

When a virtual machine runs a program, it needs memory to store many things, including bytecodes and other information it extracts from loaded symbol files, objects the program instantiates, parameters to methods, return values, local variables, and intermediate results of computations. The symbol virtual machine organizes the memory it needs to execute a program into several runtime data areas. the memory will be divided into a number of different regions to store different information, thus forming a specific memory image structure, in Figure 2 symbols on the virtual machine memory image of the description, including the overall data, symbols code, stack areas, Mapping function parameters, including global data storage area for the symbols in the overall process data; symbol of the storage area code from the symbol library in order to load the executable code symbols; stack storage area for the function of the parameters, return values, Local variables and intermediate results; another because of graphic rendering of graphics interface function by the host GIS platform, to the implementation of the callback way, the parameters of the transmission mechanism and prepare for the virtual machine does not function as local And therefore also need a mapping function parameters area to support the implementation of mapping function.

#### 3.2 Execute engine and instruction set

The execution engine is one part of the virtual machine that can vary in different implementations. On a virtual machine implemented in software, the simplest kind of execution engine just interprets the bytecodes one at a time. Another kind of execution engine, one that is faster but requires more memory, is a just-in-time compiler. In this scheme, the bytecodes of a method are compiled to native machine code the first time the method is invoked. The native machine code for the method is then cached, so it can be re-used the next time that same method is invoked. A third type of execution engine is an adaptive optimizer. In this approach, the virtual machine starts by interpreting bytecodes, but monitors the activity of the running program and identifies the most heavily used areas of code. As the program runs, the virtual machine compiles to native and optimizes just these heavily used areas. The rest of the of code, which is not heavily used, remain as bytecodes which the virtual machine continues to interpret. This adaptive optimization approach enables a virtual machine to spend typically 80 to 90% of its time executing highly optimized native code, while requiring it to compile and optimize only the 10 to 20% of the code that really matters to performance. Lastly, on a virtual machine built on top of a chip that executes bytecodes natively, the execution engine is actually embedded in the chip.

Each thread of a running program has its own pc register, or program counter, which is created when the thread is started. The pc register is one word in size, so it can hold both a native pointer and a returnAddress. As a thread executes a Java method, the pc register contains the address of the current instruction being executed by the thread. An "address" can be a native pointer or an offset from the beginning of a method's bytecodes. If a thread is executing a native method, the value of the pc register is undefined.

Every instruction consists of an opcode followed by zero or one parameters. Each opcode is one byte in size; an instruction parameter has four bytes. A few "debugging" instructions (at the end of the list) form an exception to these rules: they have two or more parameters and those parameters are not always 4 bytes size. Many instructions have implied registers as operands. This reduces the number of operands that are needed to decode an instruction and, hence, it reduces the time needed to decode an instruction. In several cases, the implied register is part of the name of the opcode. For example, PUSH.aux is the name of the opcode that stores the AUX register on the stack. This instruction has no parameters: its parameter (aux) is implied in the opcode name. Because of the need to carry out drawing operations, in addition to its support generic virtual machine instructions, there is a need to support drawing instructions. the drawing instruction should be implement as call back function.

#### 4. SYMBOL LIBRARY

The Symbol library is a binary file (on disk) contains a set of symbols information. Actually it is a program that can be load and execute by the symbol virtual machine, all the symbols in the library are described by the virtual machine's instruction (or byte code), each symbol in the symbol library has a series of instructions , possibly also contain data. The structure of the symbol library is similar to the memory image layout (shown in figure 2), it consists of the "head information", and the code and data sections, but the heap and stack section are not stored in the binary file, since the symbol virtual machine can build the from information in the "head information" section. Once the symbol library is loaded by the symbol virtual machine, it can be executed by the execute engine.

#### 5. SYMBOL DESIGN TOOLS

The symbol design tools consist in four components:

(1).symbol design language: It is a special programming language used to create symbol libraries source, called vmSymol Language. The vmSymol language is similar to the C programming language, but it exclude many complex concept of C, you can think it is a simplified C programming language. At the same time, the Language added many predefined functions for symbol design.

(2).symbol complier: It is the compiler for the vmSymbol Language. The compiler takes a source code and converts it to instruction (or bytecode), which is subsequently executed on symbol virtual machine. The compiled result is what we common called symbol library.

(3).symbol IDE(Integrated Development Environment): It refers to the vmSymbol IDE, including the source editor, the compiler, the debugger and the symbol preview tools. The IDE support for syntax styling, error indicators, code completion and call tips. The user can design the symbol libraries inside it very conveniently.

#### 6. APPLICATION EXAMPLE

Virtual machines based symbols system has been used in Fujian electric management and service system. In this system, there are tow places need electric map: one is the in the information management subsystem which use ArcGIS to show electric map and run on desktop computer, another is in the information collection subsystem, It's self-development embedded GIS

System run on PDA. They all use a same virtual machine based symbols. The following is the system snap:

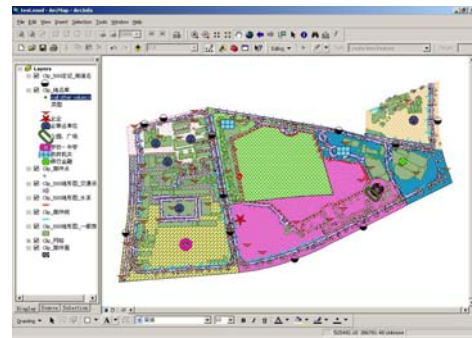


Figure 3. example in ArcGIS



Figure 4. example in PocketPC

#### 7. CONCLUSION

Virtual machine based GIS symbols system using symbol virtual machine screened the difference of the concrete GIS platform, and provide a uniform symbol interface for the GIS symbols designer, thus the symbols made by this method can be used in different GIS platforms, and achieve the same representation results. Practice proves that this is a practical solution.

#### REFERENCES

- [1][http://atlas.nrcan.gc.ca/site/english/learningresources/carto\\_corner/map\\_content\\_carto\\_symbology.html](http://atlas.nrcan.gc.ca/site/english/learningresources/carto_corner/map_content_carto_symbology.html)
- [2]SuperMap Vector Map Symbol Library Exchange Format 2.0, Beijing Super Map GIS Lit. 1999.12. <http://www.gischina.com/maindoc/simchin/gisforum/format/vector006.htm>, (Chinese)
- [3]Tim Lindholm, Frank Yellin, Java(TM) Virtual Machine Specification, The (2nd Edition) (The Java Series)[M],1999,p1-1034
- [4]Bill Blunden, Virtual Machine Design and Implementation in C/C++[M],Wordware Publishing,2002:p1-296

- [5] Philip J. Koopman, Stack Computers, [http://www.ece.cmu.edu/~koopman/stack\\_computers/index.html](http://www.ece.cmu.edu/~koopman/stack_computers/index.html) [EB], 2008-4-15
- [6] Zhong Ershung, etc, Comment of GIS Multi-source Data Integrate Model. [C], Proceedings of Geoinformatics 99?Conference, Shenzhen, 1999.8, pp286-290,(Chinese)
- [7] Song Guanfu, etc., Research for Seamless Integration of Multi-source Spatialdata, [C], Proceedings of Geoinformatics 99?Conference, Shenzhen, 1999.8, pp372-377, (Chinese)  
Wang Jun, etc, Chinese National Standard Electric Map Symbol System, 2002, (Chinese)
- [8] Ma Yaofeng, A STUDY ON THE SYMBOLIC CONSITITUTION ELEMENTS AND ITS DESIGNING MODELS, [J]. ACTA GEODAETICA et CARTOGRAPHICA SINICA, 1995.11, Vol.24, No.4. pp309-315. (Chinese)
- [9] Bai Dianyi, New Trend of Map Symbol Standardization, [J], Chinese Standardization, 1997, No.8, (Chinese)  
Nin Sanjin, Trend of Map Symbol, [J], MAP, 1997, No.1, (Chinese),
- [10] Wang Wei, etc. Designing and Implementation of Map Symbol Database Based on COM. [J], GEOMATICS AND INFORMATION SCIENCE OF WUHAN UNIVERSITY, 2002.Vol.27 No.3, pp296-230, (Chinese)
- [11] Zhao Hongjiang, Yin Ganhua, The Design and Realization of all-purpose Map Symbol Base, [J], Bulletin of Surveying and Mapping, 2002, No.4. (Chinese)
- [12] Yang Chuncheng, etc., Designing and Implementation of Map Symbol making tool based on OOP technique, Science of Surveying & Mapping, Vol. 27, 2002(1), 50-53, (Chinese)

#### ACKNOWLEDGEMENTS

The work described in this paper is supported by National Science and Technology Supporting Program of China (No: 2006BAB10B03), and supported by Hubei Natural Science Fund for Innovation groups projects (No: 2006ABC010).