

On Forming a Library of Script Tools for Computer Network Management

Vladimir Sayenko

Kharkov National University of Radio Electronics (KhNURE)

Kharkov, Ukraine

vladimir.sayenko@nure.ua

Abstract—Some procedures for network management functions are presented. The procedures help to form a library of script software tools and support the existing library. They also help to design a new script program. The library uses a description of special information Space to classify any new script program. The elements of this Space are program units. This paper proposes the new architecture of script program. The architecture is based on the samples. All solutions admit that one of the standard frameworks is chosen in the company and is used as a basis for computer network management. Scripts use WMICIM space to determine monitoring variables. Typical samples of script-programs are proposed. These samples use Java Script and Power Shell program languages. The samples help to an administrator to design special library of such tools and organize the network management process.

I. INTRODUCTION

Network management refers to the activities, methods, procedures and tools that pertain to the operation, administration, maintenance and provisioning of Network system. There are literally hundreds of tools to help network administrators do their jobs. Nowadays the Network Management uses the special frameworks, such as MOF, FCAPS, ITIL [1, 2]. Each of them supports a set of the management functions. FCAPS is the basic concept. FCAPS declares only five functions: *configuration, performance, fault, security, accounting*. MOF 4.0 declares 16 SMF functions [3]. They are in fact the extension of the FCAPS. The concept FCAPS is independent of operating systems, but Microsoft MOF and ITIL are used for Windows OS and Linux (Unix). A company should choose one of management concept and framework before to manage the infrastructure of a computer network (Fig. 1). Next stage is a choosing a network management software to support all management functions.

One of the way to support the computer network is the usage of special software platforms. The well-known platforms are Microsoft System Center (Configuration manager [4], Operation Manager [6]), HP Management solutions [7], IBM Tivoli [7]. The disadvantage of such systems is their high cost, complexity, and secrecy. An opposite simple solution is a usage of a special set of program units. They could be organized as a library of script tools. A network administrator should do any actions only according to the management functions and the framework. The functions set actions and the actions should be done by using special software tools. So, an administrator has to use a special library of the administration tools. The main question is how to form such library.

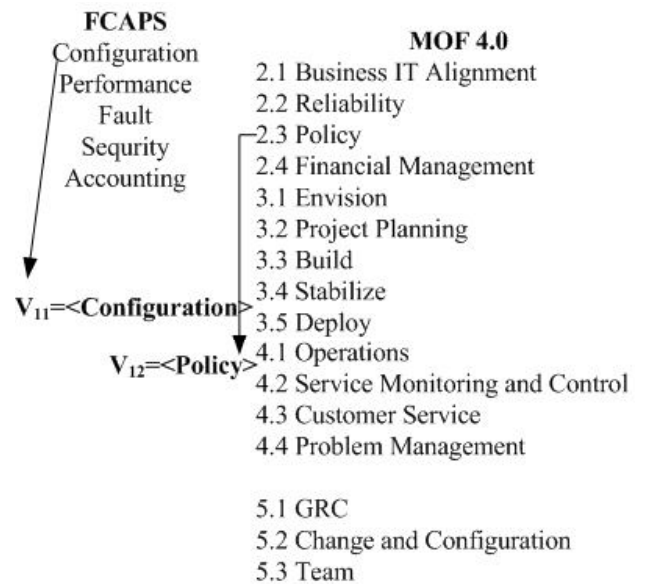


Fig. 1. Frameworks

II. CONCEPT ANALYSIS ON NETWORK MANAGEMENT

A. Software solutions

As shown in Fig. 2, there are one of five ways to release management functions in the company: a) *usage of a special management platforms* (MSSC CM, MSSC OM, HP Management solutions, IBM Tivoli), b) *usage of a network management systems* [9], c) *usage of a special programs* [10], d) *usage of a company's special programs* [11], e) *usage of scripts* [12].

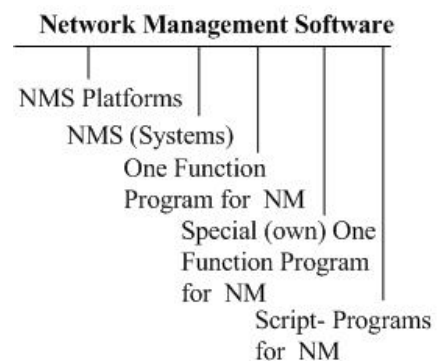


Fig. 2. Network management software solutions

No one of the ways is the best. The company's choice depends on company's financial state and some extra conditions. We will consider the usage of scripts.

B. Script solutions

As shown in Fig 3, there are four types of scripts for Windows OS environment: cmd (the packet oriented), vbs (Visual Basic), jr (Java Script), ps1 (Power Shell). The cmd\wmic is an extension of cmd.

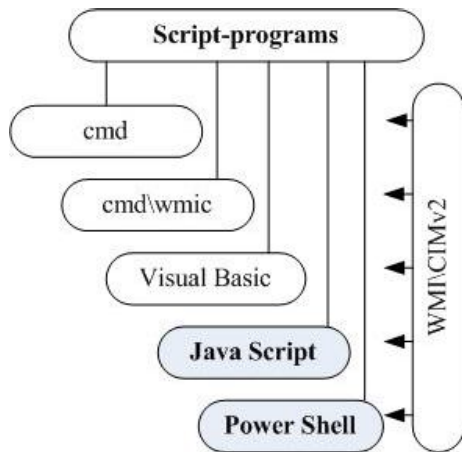


Fig. 3. Script technologies solutions

We recommend to choose Power Shell [16, 17] and Java Script [18, 19] languages. PowerShell is useful for configuration management tasks. Java Script is "lighter" than PowerShell. It utilizes less memory and is general "faster" than PowerShell at doing the same performance management tasks. In spite of restrictions on using the Active X objects and using the Internet Explorer to represent output data, the Java Script is more convenient to solve tasks for network performance area.

C. Space of monitoring variables

There are two technological solutions to compose a space of network management variables. They are SNMP [13] and WBEM\CIM or WMI\CIM [14]. Both of them have the special spaces as it shown in Fig 4.

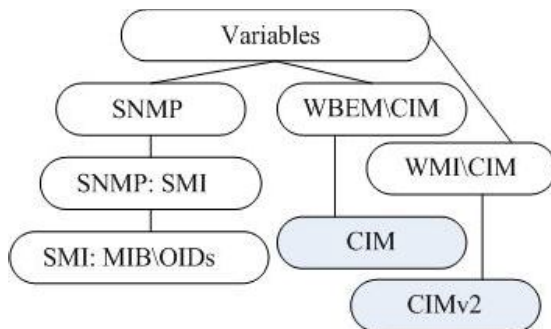


Fig. 4. Spaces for monitoring variables

The WMI\CIM and CIMv2 are more power and more preferable to choose in the extended network management solutions. WBEM\WMI [15] provides complete monitoring of

Windows operating systems – including operating system metrics, service state, process state, file system usage, and much more.

D. Monitoring modes

The management program gathers and collects monitoring information. Let there are four modes for usage of programs: on-line reporting, on-line controlling, reporting, and collection. Each of the modes requires specific reporting. They are an operational console, txt file, xls report, mdb file. As shown in Fig 5 all these solutions are convenient to use the management functions.

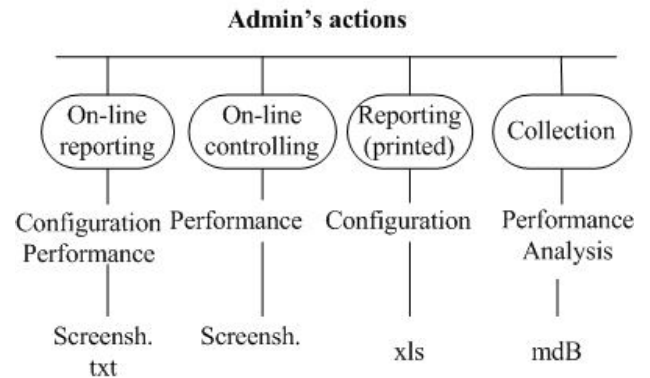


Fig. 5. Modes for usage of programs

E. Network Management Process

Fig.6 illustrates the usage of any framework for computer network management.

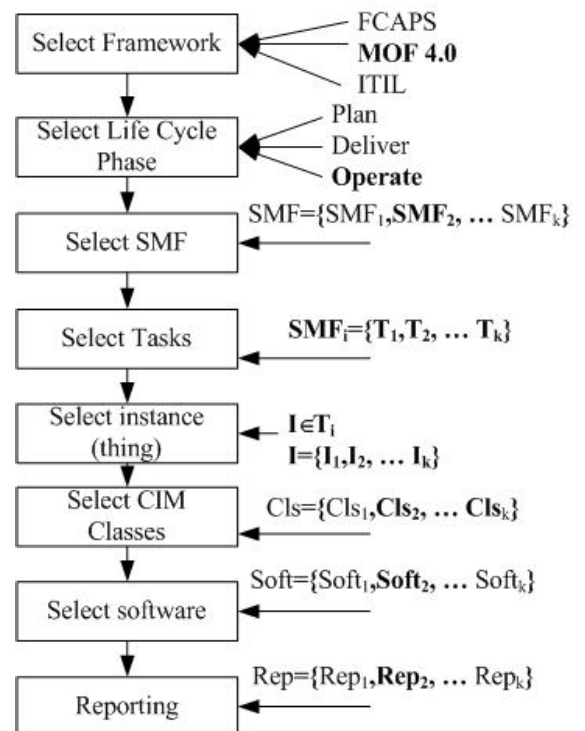


Fig. 6. Schema of network management

First of all the company should choose one of a framework. The administrator determines the life cycle phase for the computer network. He chooses SMFs and determines the set of management tasks. According to the tasks, he selects instances (managed things) and after that, it is possible to select CIM classes. At the last step, he can choose or create the corresponding software program tool and form of reporting. Administrator collects all software tools in the archive and organizes them as a special library.

III. SPACE OF SCRIPT TOOLS

To form the library of special software tools we should determine the correspondence information Spaces.

Let's describe two such spaces: *Space A*, *Space B*. The *Space A* describes a software unit $L(v_1, v_2, v_3, v_4, v_5, v_6)$ as an element of network management process accordance to Fig.5. The position of any $L(v_1, v_2, v_3, v_4, v_5, v_6)$ could be given as an element of a hypercube. The 3D hypercube is presented as shown in Fig. 7.

The hyper cube is determined as *Space A* = $(v_1, v_2, v_3, v_4, v_5, v_6)$, where $v_1 = \langle \text{framework} \rangle$, $v_2 = \langle \text{SMF} \rangle$, $v_3 = \langle \text{task} \rangle$, $v_4 = \langle \text{instance} \rangle$, $v_5 = \langle \text{mode} \rangle$, $v_6 = \langle \text{report} \rangle$.

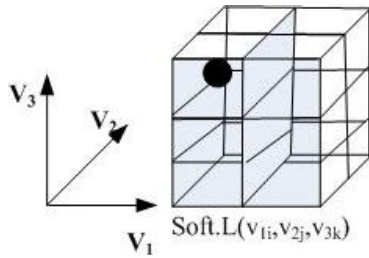


Fig. 7. 3d cube

The vector $(v_1, v_2, v_3, v_4, v_5, v_6)$ is in fact the identifier for software unit $L(v_1, v_2, v_3, v_4, v_5, v_6)$.

The *Space B* describes another hypercube. It considers a software unit as a set of small typical software samples.

Space B = (d_1, d_2, d_3, d_4) , where $d_1 = \langle \text{script language} \rangle$, $d_2 = \langle \text{mode} \rangle$, $d_3 = \langle \text{CIM class} \rangle$, $d_4 = \langle \text{report} \rangle$.

The convenience between *Space A* and *Space B* exists. We can go from *Space A* to *Space B* but not vice versa.

The *Space A* helps to order and organize all software units, and, in fact, helps to form the library. It helps to find a program if we know the management task. As additional, it helps to form the *Space B*.

The *Space B* helps to create the new software unit.

The examples of some management tasks are:

- 1) Find all software programs units for the determined type of reporting;
- 2) Find all software program units for the determined mode of a monitoring process.

IV. ARCHITECTURE OF THE PROGRAM SAMPLES

The software program architecture could be presented as a core (Script) and added modules. This architecture is used for samples of program units. Fig. 8 illustrates that the core supports access to the monitoring variables (CIM Classes). The additional modules support the functions of requesting, gathering management data and reporting.

This concept proposes the usage of samples. They are the modules for scripts.

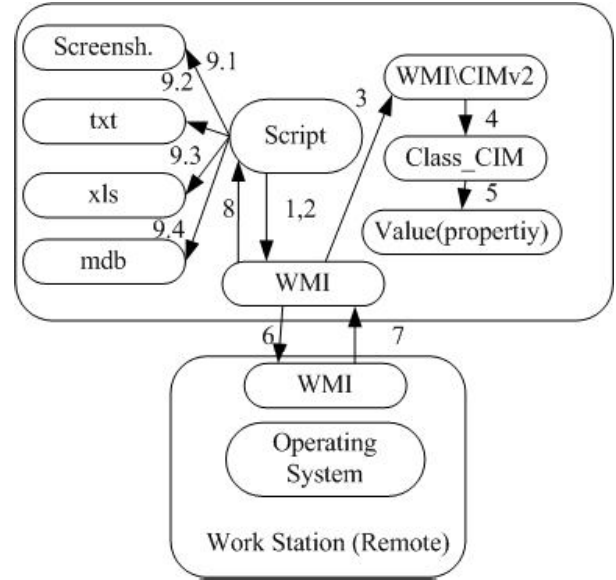


Fig. 8. Schema of software architecture

V. CLASSIFICATION PROCEDURES

To implement the proposed concepts and methods, we can use three procedures.

A. Procedure 1. Selection of the program

- 1) Describe the first management task.
- 2) Select the seek properties as values of the dimensions of *Space A* $(v^{(1)}, v^{(2)})$.
- 3) For these properties, we can seek out the script program units $L_k \in L$ and

$$L' = \{L_k \mid \forall L_k \exists (i, j), v_j = v^{(1)}, v_i = v^{(2)}\}.$$

B. Procedure 2. Demands for design a script program

- 1) If the procedure 1 is unsuccessful, it is needed to design a new program. This program will include the samples.
- 2) Determine the *Space A*.
- 3) Go to *Space B* from *Space A*.
- 4) Choose the convenient sample of the core, the module of reporting, the module of the mode, classes of monitoring variables.
- 5) Develop the script.

C. Procedure 3. Forming the library. Add an a program unit

- 1) Select the script program from Internet resources.
- 2) Rate the usefulness of the script.
- 3) Determine the variables that the script uses.
- 4) Determine the convenient CIM class.
- 5) Describe the typical task of management for the script.
- 6) Choose a framework.
- 7) Choose the SMFs.
- 8) Create classifier and identifier of the program.

VI. EXAMPLES

We created two typical samples of script-programs which the first program is been design with the windows Power shell. It shows the Configuration settings on our computer networks. We also considered the second program Java Script that has been written to show Performance properties of network processes.

The PowerShell program is getting configuration information about Win32_system Configuration and writing it on the screen. The program uses WMI\CIM object for the Win32_SystemBaseBoard WMI class in the root\CIMV2 space.

```
$strComputer = "." # "WS260-11"
$colItems = get-wmiobject -class "Win32_SystemBaseBoard"
-namespace "root\CIMV2" -computername $strComputer
$var.Description = $colItems.Description
$var.Manufacturer = $colItems.Manufacturer
$var.Product_Name = $colItems.Product
```

It shows the Configuration Data and WMI provider provides a store that is used to describe the system. The Win32_SystemBaseBoard WMI class represents the configuration of a computer system running Windows. The program additionally converts results to html which is been run off-line on the Internet browser.

The second program is designed with Java Script. It is a sample of the specific type of scripts that is convenient to use in Internet Explorer. It is used to get performance information about TCP transmitting by use WMI\CIM object for the Win32_PerfRawData_PerfProc_Process class in the root\CIMV2 space. This class provides raw data from performance counters that monitor running application program and system processes. The script displays the same data that appears in perform for the % Processor Time counter of the Process objects.

VII. DISCUSSION AND FUTURE WORK

Future work will focus on enhancing the proposed method to manage the cloud resources. The library will have the extended structure.

VIII. CONCLUSION

The main results of these investigations are the schema of management (Fig. 6), Concept of the Spaces A, B; software

architecture (Fig. 8), network management procedures (Proc. 1, Proc. 2, Proc. 3).

All solutions have been designed to help the administrators to monitor the configuration and performance tasks for workstations in the computer networks and create the special administering library of the script tools.

The practical significance of the results is that they could be successfully used to solve problems of computer networks infrastructure management

ACKNOWLEDGMENT

All solutions obtained according to the research project in "Network Management" laboratory, Control Information System department, Kharkov National University, Kharkov, Ukraine.

REFERENCES

- [1] Aleksander Clemm. *Network Management Fundamentals*. CISCO Press, 2007- 510 p.
- [2] Mani Subramanian. *Network Management: Principles and Practices (2nd Edition)*. Prentice Hall; 2012 – 726 p.
- [3] *Microsoft Operations Framework (MOF). MOF Executive Overview version 4.0*. Microsoft Accelerators, 2008 - 32 p. Web: <http://microsoft.com/technet/Solution>
- [4] Mitch Tulloch, Symon Perriman. *Introducing Microsoft System Center 2012 R2*. Microsoft Press, 2014 - 181 p.
- [5] Manoj Pal, Manish Raval. *Microsoft System Center Configuration Manager Field Experience*. Microsoft Press, 2013 - 92 p.
- [6] Danny Hermans, Uwe Sturtz, Mihai Sarbulescu. *Microsoft System Center Operations Manager Field Experience*. Microsoft Press, 2015 - 128 p.
- [7] *Network Management Solutions*. Hewlett Packard. Web: <http://www8.hp.com/us/en/software-solutions/network-management/index.html>
- [8] *Implementing a Tivoli Solution for Central Management of Large Distributed Environments*. ed. Morten Moeller, IBM Redbooks, 2005- 714 p.
- [9] Richmond Adebaye, Theophilus Owusu. *Network Systems Management 2nd Edition*. CreateSpace Independent Publishing Platform; 2013 - 478 p.
- [10] David Josephsen. *Building a Monitoring Infrastructure with Nagios*. Prentice Hall; 2007- 264 p.
- [11] Thomas A. Limoncelli, Christina J. Hogan, Strata R. Chalup. *The Practice of System and Network Administration, 2nd Edition*. Addison-Wesley Professional; 2007 – 1056 p.
- [12] Bobby Malik. *Practical Windows Administration Scripting (Administrator's Advantage)*. Charles River Media, 2006 - 424 p.
- [13] Douglas Mauro, Kevin Schmidt. *Essential SNMP, Second Edition*. O'Reilly Media; 2005 - 462 p.
- [14] *DMTF Standards: Web-Based Enterprise Management, Common Information Model, Desktop and Mobile Architecture for System Hardware* General Books LLC/ 2010 – 40 p.
- [15] Ed Wilson. *Microsoft Windows scripting with WMI: self-paced learning guide*. Microsoft Press, 2005 – 370p
- [16] Richard Siddaway. *PowerShell and WMI*. Manning Publications, 2012 - 552 p.
- [17] Bruce Payette. *Windows PowerShell in Action, 2nd Edition*. Manning Publications, 2011 - 1016 p.
- [18] Eric T. Freeman, Elisabeth Robson. *Head First JavaScript Programming*. O'Reilly Media, 2014 - 704 p.
- [19] Brandon Atkinson. *Custom SharePoint Solutions with HTML and JavaScript: For SharePoint 2013 and SharePoint Online*. Apress, 2015 - 252 p.