

Open source integrated remote systems and network management with OpenRSM

M. Kalochristianakis, E. A. Varvarigos
Department of Computer Engineering and Informatics
University of Patras
Rion, Greece
manos@ceid.upatras.gr

E. A. Varvarigos
Networks Technology Sector
Research Academic Computer Technology Institute
Rion, Greece
manos@ceid.upatras.gr

Abstract — Managing corporate Information Technology (IT) environment becomes increasingly complex as server logic architecture becomes distributed and the number of manageable entities increases. At the same time, the open source community has not yet produced a reliable systems and network management solution, even though there are open source initiatives specializing in individual fields of remote management. This paper presents OpenRSM, an integrated remote management system created by integrating individual open source initiatives and augmenting them to support additional functionality so that a lightweight integrated systems and network management solution is produced.

Keywords: distributed systems management, inventory, asset management, software delivery, network management, remote desktop control, open source software

I. INTRODUCTION

OpenRSM is a pioneering initiative aiming to be an open source, lightweight solution in the field of Enterprise Management Systems (EMS). The initiative has been based on the observation that most of the components comprising an open source EMS tool can be developed by extending existing rated components offered by the open source community.

The EMS systems market is dominated by well known products, released by major enterprise systems constructors that carry system-level knowledge and experience gained over years of producing systems and management platforms. The picture of the EMS market may change in the future. The exponential rise in computing power and the maturing of the technologies involved are exposing the enterpris, the enterprise software market is also opening to the open source community and so is the market of enterprise-supporting tools, such as resource management tools. The open source community is, however, still rather immature as far as remote management systems are concerned. There are only few Network Management System (NMS) projects under the Source Forge Open Source project hosting site and only a small number of them are on the road to maturity. As far as integrated systems and network management tools are concerned, there is none at the moment. The state of the European open source management tools market is depicted in the “catalogue of available Open Source tools for the Public Administration” [1], where only a couple of network management systems appear in the section of administration and management tools.

II. RELATED WORD

Tivoli [2] by IBM, Unicenter [3] by CA, the BMC management tools [4] and the HP management platforms are the most well known high-end EMSs. The main open source competition is in the field of network management mainly, by Nagios [5], OpenNMS [6] and Hyperic HQ [7]. Groundwork [8], BigSister [9], Iteraplan [10] are also valuable systems in the field of open systems and network management.

It is difficult to picture an EMS without full network management support. Nagios, OpenNMS, ZenoSS core, GroundWork Monitor and Hypric HQ are valuable network management platforms. The Nagios system is tested, reliable and popular and offers full monitoring and presentation functionality. OpenNMS, ZenoSS core and Hyperic HQ are also valuable systems that offer enterprise level monitoring, event management and reporting. GroungWork Monitor is a valuable project that integrates systems such as Nagios, JBOSS portal [11] and RRDTTool [12]. All the systems mentioned above offer SNMP data collection and information management, events and notification, maps and graphs. It is notable that some network management systems have incorporated system management features.

Inventory and asset management is an essential component of integrated network and systems management systems, present in all commercial high-end EMSs. OpenAudit [13], OCS [14] and H-Inventory [15] are inventory and asset management tools that take advantage of management protocols such as the ones issued by DMTF [16] in order to manage detailed information about workstations or networked devices. They rely on agents to read and send infrastructure information to databases that serve as backend for web applications that expose it to users. OpenAudit is simple, concise and organized. OCS supports load management and flexible architectures. H-Inventory is designed for dynamic environments where stations can remain offline for long periods. The open systems in this category specialize on infrastructure information reporting and do not support active system management features.

QUATOR, Arusha [17], CfEngine [18], PIKT [19] and BCFG2 [20] fall in the category of configuration management tools that formalize the settings of systems and provide configuration storing and recovery functionality. QUATOR is a mature UNIX project that uses differential logic between

current and desired configurations and also supports grid technologies. The Arusha project describes configuration information using XML and then uses it to replicate machine states. The system can be exploited by other tools such as CfEngine or PIKT. The BCFG2 tool uses visualizations and reporting to aid configuration descriptions and tasks. TALOS, OPSI, WinGET, CURL, PatchMonkey and MINST also fall in the category of software distribution tools. All the above tools are self-sufficient and do not offer integrated systems management features.

TightVNC and UltraVNC [21] are remote desktop tools based on the open protocol VNC that allows users to remotely control the desktop of systems using frame buffering. Remote desktop control is an essential characteristic of EMS. The above tools are well known, valuable, specialized and self-sufficient. Iteraplan is one of the few open source enterprise architecture management systems [22]. It maps, monitors, presents and analyzes the software and systems that are deployed across the architecture of an enterprise, aiming to optimize their use and minimize costs.

Most of the systems described above implement specific management uses within the scope of an EMS. Some network management systems such as BigSister and ZenOSS, tend to adopt basic remote command execution, the separating border between NMS and EMS. OpenRSM integrates several systems, some of which are mentioned above, that is, OpenAudit, TightVNC, Win-GET, CURL, OpenNMS in order to offer inventory, remote control, network management and remote command execution. It is designed to be extendable to include different kinds of devices under its management and customizable in order to adapt to differentiated requirements such as the ones for wireless routers that run on embedded processors. The remainder of this paper is organized as follows. In Section 2 we present the OpenRSM system architecture and components. In Section 4 describe the scalability tests of the system. Finally, in Section 5 we discuss future work.

III. THE OPENRSM SYSTEM

A. Component architecture

OpenRSM adopts a distributed agent architecture, presented in figure 1. The central control point is the OpenRSM server that manages the agents. End-users connect to the server via the management interface, a graphical user interface application. The server schedules and synchronizes the execution of jobs as defined by users. Jobs correspond to distinct administrative tasks manageable within the system. They play central role in terms of usability, design efficiency and system scalability. They implement basic system classes of service that is, inventory, remote control, remote command, etc, and once instantiated by user ,they become active and reusable. Jobs are manageable at the management console of the system. They are created graphically and they are submitted to the OpenRSM server that assigns to OpenRSM agents at the managed stations where they are decoded and executed. The OpenRSM server checks the syntax of the submitted jobs, schedules the execution, prepares the agents for job execution, concentrates reports, logs and responds back to the management console. The backend of the OpenRSM server

consists of web server(s), database server(s) and proxy server(s).

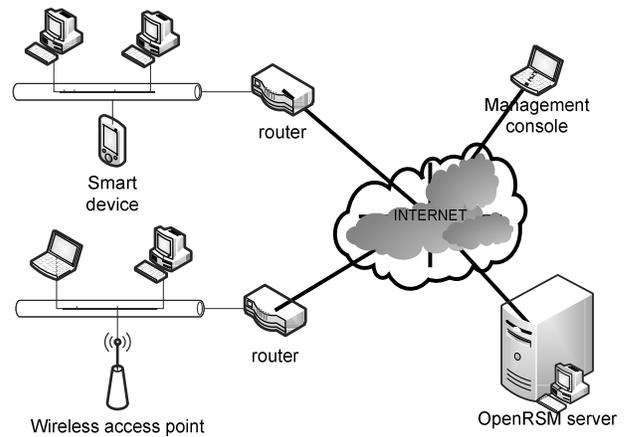


Figure 1: OpenRSM architecture and components.

B. Layers of functionality

The functional layering of OpenRSM consists of the four basic EMS subsystems that is, inventory, deployment and software distribution, network monitoring, and remote control [23]. Each subsystem is implemented by a valuable open source system and all of them have been integrated in order to interoperate in a complementary fashion.

Inventory and asset management is the core of remote systems management providing organizations with the ability to gather information on the hardware and software of the managed workstations. The technologies used, namely CIM/WBEM [24], are mature enough to provide vendor interoperability. The OpenRSM platform uses the OpenAudit inventory software designed to read assets/inventory information, store it into a database and exposed it via the web. The OpenRSM inventory subsystem builds upon OpenAudit by enhancing both automation of use and functionality. The audit software is integrated with the OpenRSM agent and can be run from a remote location in order to produce inventory audit reports. At the server side, the database schema of OpenAudit is integrated with the OpenRSM database, making asset management information available to all parts of OpenRSM. For instance, the dynamic groups feature takes advantage of this fact and offers to administrators the capability to create groups of workstations that share common characteristics stored in the inventory database.

The OpenRSM software delivery subsystem can be used to install or uninstall software packages to workstation equipped with the OpenRSM Agent. This subsystem is fully automated and complete in supporting the installation and uninstallation of software on one or more workstations. It can support all types of installation such silent ones, where no user interaction is required, if it is supported by the application installer. Silent installations/uninstallations are very useful especially for the purposes of routine administration. The OpenRSM delivery

subsystem can utilize the software name, installer options, the installer's location in the disk or URL, the uninstaller's location, etc. In order to achieve full automation OpenRSM extended the Windows-get [25] open source tool to support software uninstallation, implemented by calling the application's uninstaller and by providing it with proper information about the type of uninstallation to be performed (eg. silent or custom). Furthermore, handling archives was added in order to install software stored in them. Another enhancement was the introduction of the use of the Curl library [26] for handling file transfers in order to achieve full control over the delivery procedure thus allowing for features such as resuming broken deliveries and installation.

The OpenRSM integrates with the NINO open source Network Management System. NINO has been integrated in terms of database, web server and business logic with the rest of the OpenRSM components. Changes in the OpenRSM environment are instantly reflected in the monitoring output. NINO utilizes SNMP and WMI [27] technologies for the provisioning of rich real time monitoring information from workstations and network active elements. The list of features includes network discovery using various methods, event triggering, monitoring presets and groups, various presentation methods (web interface device browser, reports, applet graphs) and other useful features.

For the purposes of the remote desktop control subsystem, the OpenRSM uses the UltraVNC application. OpenRSM is capable of starting an UltraVNC at a managed workstation or at a group of managed workstations in order to implement reverse connections that solve the inherent synchronization problems in remote desktop control. The UltraVNC viewer is then started on the management console and connects to the managed workstation(s). If the UltraVNC server remains idle for a configurable time period, it is closed by the OpenRSM agent. Finally, the remote control service supports a number of useful features. It can be configured to ask of user permission before establishing connections and it can also be configured to use routed connections, via the OpenRSM server, instead of direct ones.

C. The agent tier

The OpenRSM agents are installed using GUI installers at the managed workstations. Their functionality is limited to the execution of commands sent by the server, and do not interact with any module of OpenRSM. Agents only communicate with trusted servers declared at installation time. They support the client systems of all the OpenRSM subsystems besides communication and subsystem integration logic. Initially, agents are in sleep mode and listen for UDP wake up packets. After they are woken they communicate with the OpenRSM server using a custom handshake where agent state and job execution information are exchanged. The execution of each job triggers communication with the appropriate server subsystem.

The OpenRSM system is capable of managing both Windows and *nix systems, through corresponding agent distributions that take into account the characteristics of each platform. Each distribution includes different agent flavors that

correspond to different types of usage; the agent can be executed as a background process for silent operation, as a graphical application user for verbose interaction, as a service, or as a console application. There is also an agent code branch that can be executed in embedded devices. Such agent flavours were used to manage MIPS based wireless routers and sensor management stations relying on TinyOS [28].

D. The management console

The OpenRSM management console, presented in figure 2, is the graphical multilingual control environment exposed to users. The console is initially used to discover the available workstations that run OpenRSM agents and are therefore manageable. The discovery mechanism is IP-based; it exploits the sleep/wake mechanisms and the server-agent communication protocol in order to retrieve the agent identification (UID), type and state and present it on the managed entities tree on the left console panel. After agents are discovered, the users can start preparing jobs and send them for execution. There are jobs for each one of the subsystems integrated in OpenRSM: inventory, remote control, software delivery and remote command execution. Users configure jobs templates that correspond to classes of the OpenRSM API and instantiate them. Once created the job instances are available through the console tree. In order to run on an OpenRSM Agent, a job must be visually combined with the corresponding managed workstation. In that case the server reads the association and sends the job to the agent. Jobs are reusable; the same job can be sent to many stations and it remains alive until it is deleted. Furthermore, both jobs and workstations can be grouped. Groups of workstations and jobs, or groups of jobs and workstations can also be combined in order to create workstation-job associations. A group of jobs can contain any kind of jobs and no dependencies are implemented. It is the user's responsibility to create a rational sequence of a group of jobs for execution. The management console provides the means for job execution supervision, reports and statistics. The job state is displayed along with information for the agent that executes it, related timestamps, etc. while filters can be applied to the job list, creating job-machines assignments that meet specific characteristics (e.g., owner, date, job type, etc).

A key usability feature of the OpenRSM console is the creation of dynamic groups of workstations or the presentation their selected attributes on a visual form. By searching in the database for the inventory results of the managed workstations, groups of machines that match specific user-defined characteristics can be created. The objective is to enable the easy identification of workstations that share common characteristics and group them together in new groups, or present their selected attributes on a visual form. The query that created the dynamic group can be executed at any time, updating the group members.

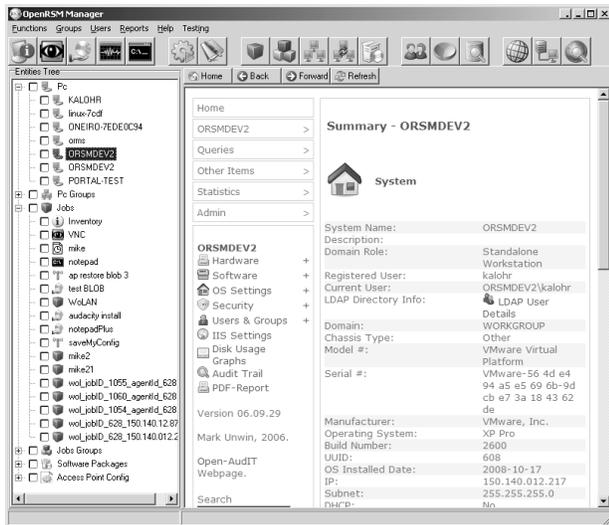


Figure 2: The OpenRSM Management Console.

The OpenRSM's reporting functionality is complemented by the data explorer form, created to provide complete database supervision. The user is capable of browsing database entities and combining their contents whenever internal linking is possible. Combinations are presented in the form of reports that can be exported in various formats, such as doc, xls, html, txt and csv. The management console is also capable of producing cumulative and detailed statistics about system utilization. The generated statistics can record general system usage, job distribution, workstation utilization and user actions. General information is presented in visual charts providing summary information on the job submission rates, job error rates and job distribution with respect to job type and submitting user. Detailed information is presented in individual reports.

IV. CONCLUSIONS AND FUTURE WORK

The OpenRSM system was successfully tested in three pilot installations in the offices of Achaia Prefecture, in the General Direction of Citizens Service Bureaus and at the Rion Citizens Services Bureau [23]. It has also been extended to include the management of wireless sensor networks and TinyOS platform services. OpenRSM was also extended in order to provide startup and shutdown management for workstations that support power management protocols such as ACPIv2 via wake on LAN jobs. Future work will enhance OpenRSM with the mechanisms that save and restore the configuration of network active elements. The implementation will be based on integrating known SNMP based tools, the appropriate MIBS and on the construction of appropriate job classes. Migration to pure web technologies such as java is also discussed along with print management and accounting solutions and bootp.

REFERENCES

- [1] Consortium for studying, evaluating and supporting the introduction of Open Source software and Open Data Standards in the Public Administration (COSPA), Deliverable 2.1 for WP2 "Catalogue of available Open Source tools for the PA", 2006
- [2] IBM tivoli software, <http://www-01.ibm.com/software/tivoli/>, July 2010
- [3] CA NSM, <http://www.ca.com/us/system-management.aspx>, July 2010
- [4] BMC management tools, <http://www.bmc.com/>, July 2010
- [5] The NAGIOS network management system, <http://www.nagios.org/>, July 2010
- [6] The OpenNMS network management system, http://www.opennms.org/wiki/Main_Page, July 2010
- [7] The HYPERIC systems management platform, <http://www.hyperic.com/>, July 2010
- [8] The Ground work management platform, <http://www.groundworkopensource.com/>, July 2010
- [9] Bigsister systems and network monitor, <http://www.bigsister.ch/>, July 2010
- [10] Iteraplan EA management, <http://www.iteraplan.de/en>, July 2010
- [11] The JBOSS portal, <http://www.jboss.org/jbossportal>, July 2010
- [12] The RRD tool homepage, <http://oss.oetiker.ch/rrdtool/>, July 2010
- [13] The OpenAudit assets management system, <http://www.open-audit.org/>, July 2010
- [14] The OCS inventory, <http://www.ocsinventory-ng.org/>, July 2010
- [15] H-Inventory, <http://sourceforge.net/projects/h-inventory/>, July 2010
- [16] Distributed Management Task Force, <http://www.dmtf.org>, July 2010
- [17] M. Holgate, W. Partain, "The Arusha Project: A Framework for Collaborative UNIX System Administration", Proceedings of the 15th USENIX conference on System administration, pp 187-198
- [18] PCFEngine, <https://info.enstb.org/projets/pcfengine/en/index.html>, July 2010
- [19] PIKT systems monitor, configuration management, <http://pikt.org/>, July 2010
- [20] BCFG2, <http://trac.mcs.anl.gov/projects/bcfg2/>, July 2010
- [21] The UltraVNC remote control software, <http://www.uvnc.com/index.html>, July 2010
- [22] Henk Jonkers, Marc M. Lankhorst, Hugo W. L. ter Doest, Farhad Arbab, Hans Bosma and Roel J. Wieringa, Enterprise architecture: Management tool and blueprint for the organization, Information Systems Frontiers, Volume 8, pages 63-66, Number 2 / February, 2006
- [23] Y. Karalis, M. N. Kalochristianakis, P. Kokkinos, E. A. Varvarigos, OpenRSM: a lightweight integrated open source remote management solution, Intl Journal of Network management, volume 19 issue 3, pages 237 - 252
- [24] WBEM, <http://www.dmtf.org/standards/wbem>, July 2010
- [25] Windows-get, <http://windows-get.sourceforge.net>, July 2010
- [26] Curl <http://curl.haxx.se>, July 2010
- [27] WMI and SNMP: <http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/feasability/wmisnmp.mspix> July 2010
- [28] The TinyOS operating system for wireless sensors, <http://www.tinyos.net/>, July 2010