SNMP vs. WBEM

The Future of Systems Management

Over the last few months, I've been involved with system architects debating the pros and cons of SNMP and WBEM. Through these discussions and my personal experience, I've decided to write up my thoughts. The bottom line of this discussion is to explain why I think WBEM is a far superior systems management technology and how it addresses the short comings of SNMP. It should also highlight the major reasons why a rich systems management specification was needed in order to fill the void in present and future complex and large scale systems management environments.

Let's start by examining the abbreviations of SNMP (Simple Network Management Protocol) and WBEM (Web-Based Enterprise Management)…

Simple

This is one of the reasons why WBEM came about. SNMP can model simple management environments. When SNMP was initially established, it was used to manage routers and other network related equipment. These types of hardware didn't require a complex management environment. In addition, the processing power of the hardware (usually) limited the scope of what it was able to run and so the management solution had to be lightweight. SNMP fitted perfectly. Routers (and the like) didn't require a rich management interface and it was also lightweight enough to run in the embedded software. Now, the horizon is very different! Systems are more complex than routers (and network related hardware) and their management interface has stretched far beyond the simple nature of SNMP. Hardware now runs on much more powerful processors with more memory which in turn means the software that runs on them is capable of much more functionality. Every year these new systems get more complex, more powerful and integrate more closely with an organisation's network infrastructure. Modern voicemail systems, for example, can directly integrate with an organisations network, directory, email server and systems management infrastructure. *Simple* doesn't meet the new requirements of large scale complex enterprise systems. It's not surprising that a new standard is required to meet this new challenge. A few years ago SNMP didn't need to cater for large scale enterprise systems management the way it is defined now. Hence the WBEM initiative was born. This is the first nail in the SNMP coffin.

Network

RFC 1157 – A Simple Network Management Protocol (SNMP), indicates that the protocol's initial requirements were to manage network hardware. Quoted from RFC 1157:

> *Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations.*

The WBEM initiative does not necessarily see hardware as being sufficiently different from software. Both hardware and software are required to have a running system. This is the second nail in the SNMP coffin.

Management

Studying the term *Management* from a WBEM perspective reveals that SNMP as a technology does not have management objects. At best, SNMP has management variables. In the WBEM world, you have namespaces which contain management classes. A management class is a definition of what management objects will look like. This includes properties and methods. Because WBEM uses object oriented principles, management classes use inheritance to communicate properties through the chain of derived classes. Most operations in the WBEM world use management objects to manage the system. However, probably the most important aspect of large scale and complex systems management is the ability to draw relationships between management classes and management objects. A simple example may be *which hard disk controller is this hard disk connected to?* SNMP's closest parallel to WBEM namespaces are called groups. SNMP does not have the concept of management objects in the way that WBEM defines them. The closest parallel of SNMP variables are WBEM management class/object properties. As SNMP does not have management classes to logically encapsulate related properties, all SNMP variables must be listed in what can best be described as a large table. SNMP has no direct equivalent of management methods. This is the third nail in the SNMP coffin.

Protocol

SNMP is a *protocol*. This has been SNMP's largest success, but it is also SNMP's largest downfall. SNMP's management information structure has its network transport protocol very closely tied to the representation of management information. This stops SNMP from significantly moving forward because of the backwards compatibility co-existence requirements it has to meet. WBEM is an *initiative*. The founders had the foresight that the management information is separate to how it gets transmitted over a network. They also recognised that the IT industry has to move fast in today's environment. So WBEM is actually a collection of standards that aid large scale systems management. Today's common network transports (i.e. HTTP) may be different to the transports in the future. At which time another transport protocol can be easily introduced in the collective of specifications. The separation between information and network transport protocols is an important distinction. The WBEM specification has already gone through this ever changing process. It was recognised that industry wanted to use XML to mark-up management information and so the Distributed Management Task Force (DMTF) defined the *CIM in XML* specification. Later, industry also required common cross-platform interoperability and so the DMTF defined a new network transport protocol called *CIM operations over HTTP*[1]. This demonstrates how the WBEM initiative will move quickly over the coming years. SNMP doesn't have this luxury. It's a protocol. It is tied-in with the original version 1 requirements when it made its debut in August 1988. SNMP's success has derived from it being a protocol. It has enjoyed much success from many common management applications developed by corporations such as HP and IBM, because there has been reasonably good heterogeneous interoperability. However, as more and more systems are added to large enterprise networks, managing those systems is closely coupled with the fact that SNMP is not discoverable. This is a serious limiting factor. If a management application receives an event (or *trap* in SNMPv1 or *inform* in SNMPv2 terminology), it cannot discover what the information means without a preconfigured system having the network transport data described by a Management Information Base (MIB) file. The same goes for SNMP variables. This enforces how closely coupled the representation of the management information is to the network protocol. Also, the SNMP protocol is encapsulated by UDP[2] – a connectionless protocol. This means that requests and response may not complete

[1] More information about *CIM operations over HTTP* can be found in the article "Putting the Web back into WBEM" by Craig Tunstall and is available from http://www.wbem.co.uk/.

[2] SNMPv1 also works with Internet Protocol (IP), OSI Connectionless Network Service (CLNS), AppleTalk Datagram Delivery Protocol (DDP), and Novell Internet Packet Exchange (IPX).

their journey. So in SNMP v2, they added an acknowledgement command to the SNMP vocabulary to improve its reliability. This is the fourth nail in the SNMP coffin.

The Future

WBEM addresses all of SNMP's short comings, plus more. Microsoft's implementation of the WBEM standard is called Windows Management Instrumentation (WMI) and is built into the Windows operating system. Through WMI, administrators can easily automate their common administration tasks through Visual Basic scripts (VBScript) into their quality process. As far as I know, there is no standard way an administrator can do this for SNMP, well, until that is, Microsoft provided support to translate SNMP management information into WBEM-based management information. The reverse is not possible. You cannot dumb down the rich WBEM information into SNMP, with the possible exception of event notification (although this may not be possible either if the event has embedded object properties).

The representation of management information under WBEM is significantly more understandable than SNMP. Here is an example reference to a WBEM management object:

```
\\MYMACHINE\root\CIMV2:Win32_UserAccount.Domain="LONDON",Name="GwynCole"
```

The above easily demonstrates which management object we are identifying. It's a user called `GwynCole` who is in a domain called `LONDON` and the management object is in the `root\CIMV2` namespace on a machine called `MYMACHINE`.

Here's an example of an SNMP reference (or object identifier) to a management variable:

```
1.3.6.1.2.1.7.1.0
```

It is much harder to understand and the information that it is referring to. When you install the MIB on the client machine, you will be able to deduce the above object identifier as:

```
iso.org.dod.internet.mgmt.mib.udp.udpInDatagrams.0
```

There's an obvious omission in the above reference. Which machine is the variable is on? The reference does provide a path to the exact variable, but in the WBEM world management objects can have multiple key properties. This capability allows for more freedom to specifically reference a management object in more complex management environments (like in the previous example which had `Domain` and `Name`).

The ability to easily reference objects in the management environment is important if it is going to be adopted by the people who ultimately care about enterprise management, third party vendors, administrators, etc... The information has to be discoverable and easy to get at without further installation requirements.

Under Windows, the entire operating system is instrumented with WBEM-based management objects. With every release of Windows, more management classes and more management events are added. The DMTF working groups are developing new versions of the Common Information Model (CIM) for UNIX and Linux operating systems. CIM is the core component of the WBEM initiative. More operating systems are including support for WBEM, like Solaris from Sun Microsystems. Some operating systems like Windows are adding additional WBEM-friendly services. For example, WMI allows you to execute SQL-like queries against the management environment.

WBEM is here to stay. You could never instrument what is already available in Windows via WMI through SNMP! SNMP even if it were stretched to its outer limits, still couldn't represent relationships, logically encapsulate properties into classes and provide SQL-like query capabilities. The DMTF roadmap (http://www.dmtf.org/about/roadmap.php) shows the developments that are continuing. CIM v2.7 has just been released and CIM v2.8 is in its preliminary stages. In June/July 2003 we'll see version 2.2 of the *CIM information in XML* standard. We'll also see version 1.2 of the *CIM operations over HTTP* standard which will include the discovery of management classes in heterogeneous environments. New standards being introduced to the WBEM initiative include an Object Constraint Language (OCL) for

unambiguously specifying constraints for management classes and objects. And an official query language specification (similar to Microsoft's WQL).

References

TCP/IP Illustrated Volume 1: The Protocols by W. Richard Stevens ISBN: 0-201-63346-9

RFC 1157 (http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1157.html)

Microsoft Platform SDK

Microsoft Developer Network (http://msdn.microsoft.com)