MOSS: Monitor de Operaciones de Sistemas SCADA

Eduardo Schorb, Lucas Schorb, Sebastián Lucas, José M. Urriza, Carlos Buckle

Facultad de Ingeniería, Departamento de Informática
Universidad Nacional de La Patagonia San Juan Bosco - Puerto Madryn, Argentina
josemurriza@unp.edu.ar, cbuckle@unpata.edu.ar

Resumen. Este trabajo presenta una solución al monitoreo de las operaciones *SCADA* que se realizan en una planta industrial. Se efectúan operaciones de supervisión del estado de los dispositivos que operan en el sistema *SCADA*. La información recolectada se centraliza, se procesa y se informa, de manera que se puedan tomar acciones correctivas por parte del equipo de mantenimiento. Este sistema fue implementado en Aluar S.A.I.C. y gran parte de los requerimientos funcionales fueron solicitados por la empresa.

Palabras Claves: *SCADA*, Monitoreo, Supervisión

1 Introducción

En las últimas dos décadas, las operaciones de las plantas industriales han incrementado notoriamente el uso de los sistemas *HMI*/*SCADA* (Human Machine Interface / Supervisory Control And Data Acquisition). Es común encontrar decenas de sistemas *SCADA* distribuidos sobre una planta industrial. Sin embargo, las computadoras industriales donde son ejecutados estos sistemas, adolecen de los mismos problemas que las computadoras personales.

Problemas comunes que se pueden mencionar son: mala refrigeración, fallos de memoria, fallos de procesos en ejecución, fallos en los discos duros, etc. En incontables situaciones, estas fallas no son detectadas por días, y hasta semanas. Esto se debe, a que la mayoría de las veces el sistema solo monitorea y la acción de control es realizada por un PLC (controlador lógico programable), o un PAC (controlador de automatización programable), o un controlador industrial, etc.

Al mismo tiempo, la gran mayoría de las aplicaciones *SCADA*, pueden ser monitoreadas remotamente informando del estado de lo supervisado, los datos de adquisición o ambos. En otras palabras, se puede realizar un *SCADA* sobre los sistemas *SCADA* de la planta industrial. Consecuentemente, es posible supervisar la salud de las computadoras y de la totalidad de los sistemas en tiempo real.

Realizar este tipo de sistema es de suma utilidad cuando el número de computadoras industriales supera varias decenas o supera la centena. Se debe tener en cuenta que, la supervisión por parte del equipo de mantenimiento requiere muchas veces de verificaciones periódicas in-situ, lo cual, en plantas con

una extensión importante, insume un tiempo considerable y, en muchos casos, sin que se presenten las fallas en el sistema.

Simultáneamente, en los sistemas *SCADA* de misión crítica, es necesario saber en un período corto de tiempo, si un equipo está por fallar, si su comportamiento tiene una tendencia a fallar, o el equipo está fallando, informando a quien corresponda para una rápida solución.

Actualmente, realizar un *SCADA* sobre los *SCADAs* en grandes plantas industriales es claramente necesario, dado que se pueden reducir tiempos de mantenimiento, prevenir fallas y sobretodo minimizar los tiempos que duran las mismas. De esta manera, es posible maximizar los recursos humanos utilizados.

En las siguientes secciones se presenta el desarrollo realizado. En la sección 2 se presenta el Marco de trabajo. En la sección 3 se presenta el agente desarrollado. En la sección 4 se presenta el desarrollo y características del *SCOM*. Posteriormente, en la sección 5 se presenta la implementación del *Sistema MOSS*. Por último en la sección 6 se presentan las conclusiones.

2 Marco de trabajo

En las plantas industriales en la actualidad, las computadoras en su gran mayoría, ejecutan sistemas operativos (*SO*) de la familia *Microsoft® Windows*. Es común encontrar Windows 2000, 2000 Server, XP, Vista, 7, 2003 y 2008, como el *SO* utilizado para ejecutar los *SCADAs*. Consecuentemente, el desarrollo se realizó para estos *SO*.

Además, *Microsoft®* viene desarrollando, desde las últimas actualizaciones del *SO* Windows NT 4, el *Windows Management Instrumentation* (*WMI* - Instrumentación para Administración de Windows). El *WMI* es la implementación de la especificación *Web-Based Enterprice Management* (*WBEM* administración de empresa basadas en web), la cual es una iniciativa de supervisión industrial, realizada por la *Distributed Management Task Force* (*DMTF* - Grupo de Trabajo para Administración Distribuida [1]). La *DMTF* desarrolló un modelo estándar, *CIM* (Common Information Model) con el fin de presentar la información de administración en un ambiente empresarial y de esta manera poder acceder, configurar, gestionar y controlar los recursos de un sistema. Consecuentemente, en los *SO* de Microsoft® es posible realizar todas las tareas antes mencionadas con la mayoría de los recursos de software de Windows.

Lamentablemente, no es posible obtener información de todos los recursos y sobre todo del hardware. Consecuentemente, fue necesario desarrollar un *agente* que determine las características de hardware del equipo supervisado, la conexión a bases de datos de los sistemas *SCADAs*, etc.

A continuación se detallan las tecnologías de software utilizada en el proceso de desarrollo.

2.1 Instrumentación para Administración de Windows

El *WMI* es una tecnología que lleva más de una década de desarrollo, es estable, bien documentada y es compatible con muchos productos fuera de Microsoft®. Además, existe una política expresa por parte de Microsoft®, que incentiva a los desarrolladores a su utilización en las aplicaciones.

A continuación se define *WMI* ([2]): "*El WMI es una infraestructura escalable para la gestión de sistemas que utiliza una interface basada en estándares, orientada a objetos, consistente y extensible. WMI provee una forma estándar para interactuar con la información del sistema de gestión y la subyacente API del WMI. Este es utilizado principalmente por los administradores y desarrolladores de aplicaciones de sistemas de gestión, para acceder y manejar la información del sistema de administración".*

Por otro lado, es posible, de una manera muy sencilla, implementar esta tecnología en Microsoft® .Net Framework. Para realizar esto, se especifica como un atributo de la clase que debe realizar la publicación *WMI*. De esta manera, es posible acceder directamente a las variables compartidas, la transmisión remota segura, la administración de permisos, etc.

2.2 .Net Framework 2.0

Por compatibilidad para las máquinas que aún utilizan el *SO* Windows 2000 se utilizó .Net Framework 2.0 y no las versiones posteriores. Además, la librería utilizada para acceder al hardware también está desarrollada en .Net Framework 2.0.

2.3 Registro de Eventos/Sucesos – Visualización

Desde Windows NT 4, los *SO* antes mencionados utilizan un registro para documentar los eventos o sucesos de ejecución de las aplicaciones, servicios, hardware, etc. Este utiliza y administra una base de datos persistente, en la cual quedan registrados los diagnósticos, fallas, ejecuciones, arranque y parada de servicios, problemas de seguridad, etc. La aplicación que permite visualizar esto es el *Visor de Eventos* (en los *SO* más antiguos *Visor de Sucesos*)

El *Visor de Eventos* es una herramienta que permite supervisar el mantenimiento de los sistemas y solucionar los problemas que surjan. Es posible crear, eliminar, y administrar registros y crear entradas y vistas personalizadas. Además, se puede determinar el tamaño máximo de la base de datos, y por ejemplo, que sobrescriba las entradas más antiguas cuando se llega al límite. Por otro lado, es posible que los usuarios y aplicaciones accedan a esta base de datos, incluso de forma remota. Por último, el mismo *Visor de Eventos* instrumenta *WMI*.

En operación normal, todos los mensajes que el *agente* genera, se envían al *Visor de Eventos* y se guardan bajo el registro MOSS-Servicio.

Cada vez que el *agente* inicia su ejecución verifica en el *Visor de Eventos* que exista un registro llamado *MOSS-Servicio*. De no ser así, crea uno nuevo con el nombre *MOSS-Servicio*, de 16Mb de tamaño, y se sobrescribe, empezando por los más antiguos de ser necesario. Además, se almacenan aquí todos los mensajes que se generan, sean de información, de advertencia o de error.

2.4 Registro de Windows

El registro de Windows (registry) es la manera estándar que utilizan las aplicaciones para guardar sus configuraciones y el *SO* sus parámetros operativos. El mismo, instrumenta *WMI* y consecuentemente es posible cambiar, incluso de forma remota, cualquier configuración.

2.5 Sistema de Administración de Operaciones Centralizado (SCOM)

El Sistema de Administración de Operaciones Centralizado (Microsoft® System Center Operations Manager – *SCOM* ([3])) es un centro de datos interplataformas desarrollado por Microsoft® para diversos *SO* e hipervisores.

El software *SCOM* provee capacidades en 3 grandes áreas, que son: servicios de administración de extremo a extremo en un centro de datos; monitoreo de servidores y estaciones de trabajo bajo Windows, Linux o Unix; incremento de la eficiencia y control de las aplicaciones y sus servicios.

Este, utiliza una interface para presentar la información de monitoreo que puede ser en modo aplicación o modo web. De esta manera los administradores pueden acceder a información que les muestra el estado, la salud, el rendimiento, de las máquinas y aplicaciones que se encuentran en su dominio. Por otro lado, genera alertas para que los operadores ágilmente puedan determinar y solucionar los problemas presentados. También, es posible automatizar operaciones de reparación en caso de presentarse fallas.

2.6 Monitor de Hardware

Para obtener un monitoreo preciso de la salud de la máquina que opera el sistema *SCADA*, es necesario el sensado del hardware de la misma. Este sensado debe obtener un conjunto de parámetros que permitan determinar el correcto funcionamiento y, si este no es el ideal, generar una alarma o aviso.

Lamentablemente, no existe un estándar acerca de como acceder a la información del hardware en la industria. Cada fabricante implementa de manera diferente, la lectura de los datos.

El *Open Hardware Monitor* ([4]), es un software de código abierto y libre que monitorea los sensores de temperatura, las revoluciones de los ventiladores de refrigeración, los voltajes de la fuente, la carga de computo instantánea, la velocidad del reloj del procesador, etc. Este lo realiza monitoreando los chipsets que disponen las placas madre, placas de video, discos duros, etc.

Esta organización dispone de un programa y una librería que utiliza para acceder al hardware desde Windows de una manera transparente y homogénea. Esta librería fue la utilizada para el desarrollo del *agente*.

3 Agente MOSS

Para monitorear las operaciones que se realizan en las máquinas, se desarrollo un *agente* que se instala como un servicio de Windows del tipo demonio. De esta forma es posible acceder a la información del hardware de manera más sencilla.

3.1 Funcionalidades del Agente MOSS

El *agente* posee las siguientes funcionalidades:

- Hardware y Sensores: Expone información *WMI* por cada elemento de hardware o sensor detectado con la librería *Open Hardwate Monitor*.
- S.M.A.R.T. (Self Monitoring Analysis and Reporting Technology): Expone información *WMI* sobre los parámetros operativos del los discos duros de la computadora monitoreada.
- Información del Sistema: Expone información *WMI*, sobre los parámetros de funcionamiento de la máquina monitoreada. Estos son:
  - Porcentaje total de uso de CPU
  - Máxima temperatura de todos los cores del CPU
  - Porcentaje y total de uso de la memoria utilizada en MBytes
  - Total de memoria del sistema en MBytes
  - Cantidad de discos rígidos detectados en el sistema
  - Porcentaje y total de uso del disco
  - Porcentaje de uso del ancho de banda de red y Cantidad de tarjetas de red activas.
  - Cantidad de actualizaciones *WMI* que se llevan realizadas y fecha y hora de la última actualización.
- iFIX ([5]): Puede conectarse a través del *OBDC* con la base de datos del iFIX de la máquina monitoreada y exponer la consulta realizada a través del *WMI*. (iFIX: es un software para monitorización y control de supervisión muy utilizado en la industria para aplicaciones *HMI*/*SCADA*).

3.2 Configuraciones del Agente

Existen numerosas y detalladas configuraciones del *agente* las cuales exceden el tratamiento en el presente trabajo. Consecuentemente, sólo se realizará una breve reseña de algunas de ellas.

Todas las configuraciones del *agente* se realizan a través del Registro de Windows bajo la clave HKEY_LOCAL_MACHINE\SOFTWARE\AluarMOSS.

El *agente* lee todas sus configuraciones desde esta clave cuando se inicia el servicio. Algunas de las configuraciones posibles del *agente*, son: tiempo de escritura en el registro de sucesos; tiempo entre consultas del SMART; tiempo entre consultas a la Base de Datos iFIX; la cadena de consulta *SQL* para recuperar datos de la Base de Datos del iFIX . Además, la habilitación de la publicación *WMI* de uso de: Porcentaje de uso del CPU, Memoria, Discos, Red y uso de memoria y procesador por parte del *agente*.

El *agente* incluye en su código, validación para todas las configuraciones posibles. De esta manera, se evita desencadenar un funcionamiento erróneo por falta de una, varias o todas las claves de configuración. Si es este el caso, se usan valores predeterminados embebidos en el código y además, se genera una entrada de error en el Visor de Sucesos local de la máquina monitoreada.

3.3 Auto-diagnóstico

El *agente*, posee capacidades de auto-diagnóstico. Estas capacidades de auto-diagnóstico permiten determinar si el *agente* está funcionando correctamente en la máquina monitoreada. Consecuentemente, expone información *WMI* sobre: tiempo en µSeg entre el inicio y el final de cada ronda del *agente*; tiempo en µSeg entre el inicio y el final de la ejecución del S.M.A.R.T; tiempo en µSeg entre el inicio y el final de cada consulta al iFIX; si el proceso iFIX está en ejecución; Fecha y hora de la última consulta al iFIX; cantidad de registro devueltos en la última consulta al iFIX; cantidad de KBytes de memoria que el *agente* está utilizando; porcentaje del uso total de CPU del *agente* y cantidad de actualizaciones *WMI* que el *agente* lleva realizadas.

4 Implementación de Microsoft® SCOM

A continuación se presentarán algunas de las características de *SCOM* extraídas de [6] y algunas descripciones de su implementación.

4.1 Explorador de Salud y Grupos

El explorador de salud, es la estructura que permite visualizar, consultar y administrar alertas, cambios de estado, y otros problemas significativos generados por los dispositivos monitoreados en la red.

Los grupos son un conjunto de dispositivos administrados que pueden ser utilizados para definir el ámbito de las vistas, de los monitores y de las reglas entre otras. La creación de grupos se basó en la organización de Aluar S.A.I.C. (Figura 1).

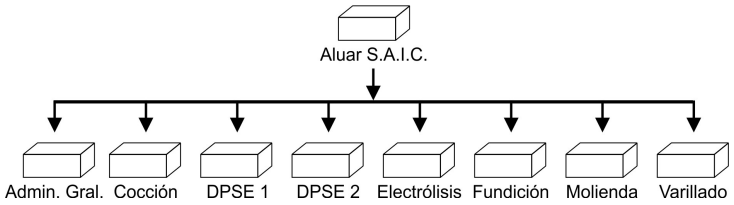

Fig. 1. Análogo a la estructura jerárquica actual.

4.2 Monitores

El sistema *SCOM* define un conjunto de grupos a los cuales se les puede asociar monitores. Estos monitores permiten supervisar el estado de elementos que conforman los dispositivos administrados. La implementación consistió en el desarrollo de monitores asociados a elementos específicos de los equipos comprendidos en los *SCADAs*.

En *SCOM* se define a los monitores como una máquina de estados finita. Cada estado se mapea a un estado de salud que permite definir condiciones de alertas. Este posee una vasta colección de tipos de monitores, los cuales tienen aplicación sobre diversos elementos de análisis. Para llevar a cabo la implementación del proyecto se utilizaron un subconjunto de dichos tipos de monitores, los cuales se mencionan a continuación.

Monitor - Contador de Rendimiento WMI.

El tipo *Contador de Rendimiento WMI* (*CRWMI*) está basado en un modelo de acceso a datos a través de consultas realizadas al *WMI* utilizando el lenguaje *WQL* (*Windows Management Instrumentation Query Language*). Este tipo de monitor, fue utilizado para la implementación del proyecto, debido a que el *agente* desarrollado es un proveedor *WMI*, que publica información de rendimiento del sistema supervisado. Las alertas generadas por los monitores, basados en *CRWMI*, están definidas por umbrales que los datos recuperados deben, o no, alcanzar.

Monitor de umbral estático.

Cuando el objeto de rendimiento monitoreado excede un umbral definido, el estado del monitor cambia. Existen dos tipos de umbrales estáticos:

- Umbral Único: este posee un límite, cuando el contador de rendimiento excede, el estado cambia. Puede configurarse para que sea *correcto*, *advertencia* o *crítico*. Los diferentes tipos de umbral único utilizados fueron:
  - Umbral simple: mide el valor de un objeto con relación a un límite establecido, de esta forma se supervisó el porcentaje de carga de CPU.
  - Muestras consecutivas en umbral: este tipo de monitor se utilizo para recuperar datos de rendimiento sobre el porcentaje de uso de memoria. El objetivo fue evitar las falsas alarmas, debido a cambios repentinos del contador.

- Umbral Doble: a este tipo de monitor se le pueden definir dos límites que el contador de rendimiento puede alcanzar. Además, puede configurarse para que el estado del monitor sea *correcto, advertencia* o *crítico*. Se utilizó para supervisar el porcentaje de uso de disco.

Plantilla Monitor de Procesos.

Esta plantilla permite detectar si un proceso está siendo ejecutado, o no, y generar una alarma en consecuencia. Se implemento para supervisar la ejecución de los procesos de iFIX.

Plantilla Monitor de Servicio de Windows.

Esta plantilla genera una alarma en el caso de que un determinado servicio no se encuentre en ejecución. Si el servicio no se encuentra activo, *SCOM* también provee una interfaz que permite iniciar remotamente dicho servicio. Está plantilla se utilizó para certificar la ejecución del *agente*.

4.3 Reglas

Las reglas son elementos que se utilizan para recopilar los datos de rendimiento de un dispositivo. Una regla no puede establecer un estado de salud. Se definieron reglas para los sensores de temperatura y de porcentaje de uso de CPU, publicados por el *agente*. Además, se crearon reglas de colección, basadas en eventos *WMI*, que recopilan los datos periódicamente y los grafica.

4.4 Vistas

Las vistas pueden personalizarse y son accesibles a través de un portal web provisto por *SCOM*. Estas, permiten visualizar un conjunto filtrado de datos, por ejemplo, un conjunto de dispositivos administrados que posean una cierta particularidad en común. Existen varios tipos de vistas como: la vista de que permite mostrar las alarmas activas de los monitores implementados, para supervisar sensores de CPU, memoria, disco; la vista de estado que permite de los dispositivos administrados y así detectar si existe una falla; la vista de rendimiento que presenta los datos de rendimiento recopilados por una regla a través de una gráfica, por ejemplo porcentaje de CPU, Temperatura de CPU, uso de memoria, uso de disco, etc; la vista de panel que reúne las vistas de estado asociadas a los diferentes grupos creados.

4.5 Paquetes de Administración

Un paquete de administración (Management Pack) es un contenedor que permite trasladar todo el conjunto de objetos creados y configurados (grupos, monitores, reglas, vistas, etc.) desde la estación de desarrollo a la organización.

En nuestro caso, los monitores, vistas y reglas creados, fueron transportados a la empresa Aluar S.A.I.C. a través de un paquete de administración.

5 Implementación Sistema MOSS

Actualmente, se administran con el *Sistema MOSS* alrededor de 150 máquinas *SCADAs*, vinculadas por una red local. En la Figura 2 se presenta un esquema de componentes de la solución expuesta.

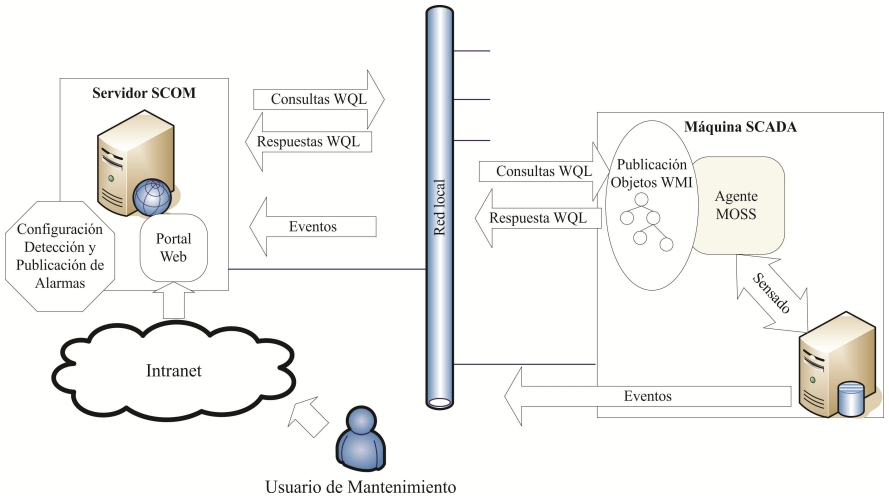

Fig. 2. Resumen de arquitectura de la Solución MOSS

A continuación se presenta un resumen de la implementación:

- Cada máquina *SCADA* tiene instalado como servicio el *agente*, con configuraciones particulares de acuerdo a la arquitectura del hardware, del sistema operativo y a las funciones *SCADA* que desarrolla.
- El *agente* realiza el sensado del hardware, del software y del registro de sucesos. Además, publica el resultado como objetos *WMI*.
- Las máquinas *SCADA* también publican eventos, a los cuales el servidor *SCOM* se puede suscribir, por ejemplo para realizar monitoreo de procesos.
- En el servidor *SCOM* están configurados:
  - Las máquinas *SCADA* están organizadas por sectores.
  - Los monitores de unidad y las reglas, que consultan periódicamente las publicaciones *WMI* de las máquinas *SCADA*, se realiza utilizando el lenguaje de consulta *WQL*.
  - Las vistas se pueden visualizar a través de la consola o del portal web.
  - Diferentes tipos de usuarios con distintos tipos de privilegios de administración, pueden consultar los dispositivos administrados, asignar mantenimiento y resolver alarmas.

Con el *Sistema MOSS* implementado, el equipo de mantenimiento supervisa de forma centralizada la información relacionada al estado de salud de los dispositivos *SCADA* y puede accionar de manera preventiva, predictiva, proactiva.

6 Conclusiones

El problema a resolver admitía varias soluciones posibles. Se eligió una solución basada en estándares y se adoptaron herramientas de plataformas Microsoft®. Se relevaron diferentes maneras de acceder a sensores de hardware, considerando que pueden diferir de acuerdo a la arquitectura y el fabricante. Se decidió utilizar una librería de uso público que permite acceso transparente al hardware. La solución del *Sistema MOSS* fue verificada en un ambiente real en la planta Aluar S.A.I.C.

Referencias.

[1] I. Distributed Management Task Force. (2012, *DMTF, Distributed Management Task Force*. Available: http://www.dmtf.org/

[2] Microsoft®. (2012, *WMI, Windows Management Instrumentation*. Available: http://technet.microsoft.com/es-es/library/cc787057%28v=ws.10%29

[3] Microsoft®. (2012, *SCOM, System Center Operations Manager*. Available: http://www.microsoft.com/systemcenter/es/es/operations-manager.aspx

[4] O. H. Monitor. (2012, *Open Hardware Monitor*. Available: http://openhardwaremonitor.org/

[5] G. I. Platforms. (2012, *iFIX*. Available: http://www.ge-ip.com/es/products/family/proficy_hmiscada_ifix/

[6] L. Poggemeyer. (2009). *Guía del usuario de operaciones de Operations Manager 2007 R2*.