

Desarrollo de un Sistema SCADA para una Columna de Destilación Benceno-Tolueno

Oscar Enrique Blanco, Marilyn Johanna Inciarte, José Andrickson Mora

Resumen—En el ámbito industrial es necesario que los sistemas SCADA sean flexibles, robustos, y se puedan comunicar con diversos dispositivos (equipos finales de control). De aquí radica la importancia de que estos sean sistemas abiertos y escalables, Por esta razón se ha desarrollado el SCADA (*Supervisory Control And Data Acquisition*) para una columna de destilación de Benceno-Tolueno, que permita supervisar las variables de importancia y controlar las variables críticas de manera remota en el proceso, con este se pueden observar las alarmas generadas en tiempo real, el comportamiento de las señales de interés a través del tiempo y almacenar los valores en una base de datos. El proceso de destilación se simuló matemáticamente utilizando el *Toolbox Simulink* de *MATLAB* y el SCADA se implementó con el software de *National Instruments LabVIEW*. Este puede ser ejecutado bajo la topología cliente-servidor, lo que permite realizar la supervisión remotamente utilizando una red.

Palabras claves—Columna de destilación, Sistema SCADA, Supervisión.

I. INTRODUCCIÓN

En la actualidad los procesos industriales han ido evolucionando de una manera tal, que pueden ser supervisados y controlados de forma remota y automática según el grado de inteligencia del sistema, en tiempo real, sin poner en riesgo a los operadores. Los sistemas SCADA [1], se crean para alcanzar este objetivo, estos comprenden una serie de aplicaciones de software, en la que se toman datos provenientes de un determinado proceso y se evalúan para determinar si están dentro de los valores deseados o si es necesario realizar alguna acción correctiva de control.

Los sistemas SCADA utilizan diversas arquitecturas de comunicación [2], una de las más básicas, es la que utiliza un servidor centralizado y uno o varios clientes, PLC (*Programmable Logic Controller*), RTU (*Remote Terminal Unit*), PAC (*Programmable Automation Controller*).

En este artículo se desarrolla un sistema SCADA para

Artículo recibido el 11 de Diciembre de 2009.

O.E.B. y M.J.I. están con la Universidad de Los Andes, Sector La Hechicera, Facultad de Ingeniería, Escuela de Ingeniería Eléctrica, Mérida, Estado Mérida, Venezuela, Tlf. +58-416-5730019, +58-424-6310537, E-mail: oscarblanco@ula.ve, marilin@ula.ve.

J.A.M. está con la Universidad Nacional Experimental del Táchira, Avenida Universidad, Sector Paramillo, San Cristóbal. Estado Táchira, Venezuela, Tlf. +58-416-8739460, E-mail: andrick@unet.edu.ve

monitorear, supervisar, y controlar una columna de destilación de Benceno-Tolueno [3]. Como primer punto se hará una descripción del proceso, el cual será simulado con el *Toolbox Simulink* de *MATLAB*. Seguidamente se describirá el desarrollo del sistema SCADA.

El esquema general se muestra en la Figura 1, este consta de una serie de dispositivos de campo, una unidad de control y una interfaz de usuario. El proceso se simulará con el software *LabVIEW* de *National Instruments* [4,5]. Este sistema es capaz de supervisar las variables del proceso, controlarlas, generar alarmas, almacenar la información en una base de datos, generar los histogramas y comunicarse en red utilizando direccionamiento por IP con otros computadores para ejecutar todo esto de forma remota, sin importar la ubicación física. Por último se extraen las conclusiones más importantes.

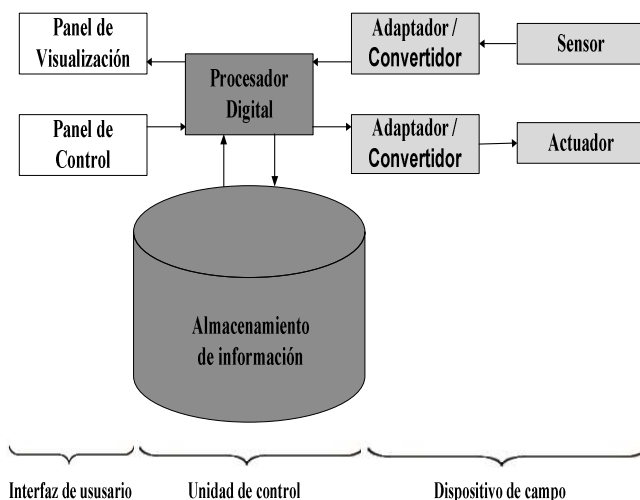


FIG 1. Esquema básico de un sistema SCADA

II. DESCRIPCIÓN DE UNA COLUMNA DE DESTILACIÓN BENECNO-TOLUENO UTILIZANDO EL *TOOLBOX* DE *MATLAB SIMULINK*

El proceso en estudio consta de un tambor de evaporación el cual se puede observar en la Figura 2, este será utilizado para vaporizar parcialmente una mezcla de Benceno-Tolueno, es decir, se desea obtener una concentración mayor del componente más volátil (Benceno) en el tope del tambor.

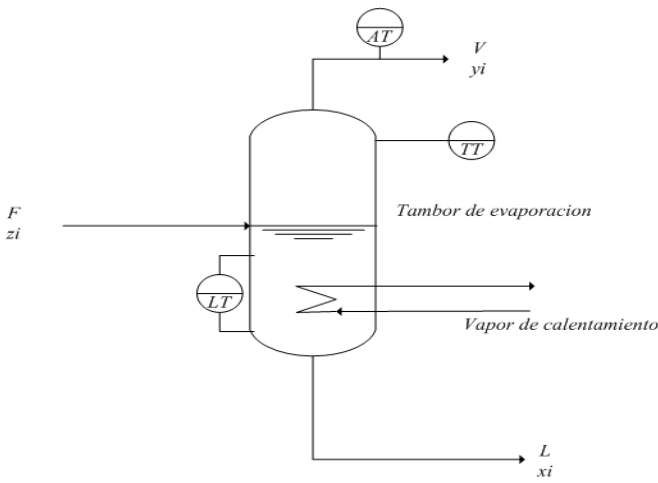


FIG 2. Modelo del tambor de evaporación de la mezcla

La alimentación del tambor (F) entra como líquido saturado con una composición del 40% molar en benceno (z_i) y la temperatura de ésta corresponde a la temperatura del punto de burbuja, en el cual se forma la primera burbuja de vapor. El proceso se desarrolla a una presión constante de 1 atm y los compuestos presentan un comportamiento ideal y se encuentran en equilibrio dentro del tambor, es decir tenemos un equilibrio de líquido-vapor.

Para calcular las condiciones en estado estacionario se realizó un balance de masa general y un balance de masa por componentes para el benceno [3], obteniéndose las siguientes ecuaciones:

$$\begin{aligned} \bar{F} - \bar{L} - \bar{V} &= 0 & (1) \\ \bar{F}\bar{z}_1 - \bar{L}\bar{x}_B - \bar{V}\bar{y}_B &= 0 & (2) \end{aligned}$$

Donde \bar{L} representa la cantidad de líquido a la salida del tanque, \bar{V} es la cantidad de vapor a la salida del tanque, \bar{z}_1 se refiere a la composición molar en benceno, \bar{x}_B describe la concentración de líquido de benceno dentro del tambor, \bar{y}_B simboliza la concentración de vapor de benceno dentro del tambor.

Al realizar un balance de masa general y un balance de masa por componentes en el benceno, ambos en estado dinámico, dando como resultado lo siguiente:

$$F(t) - L(t) - V(t) = \frac{d}{dt} \frac{M(t)}{PM(t)} \quad (3)$$

$$F(t)z_1(t) - L(t)x_B(t) - V(t)y_B(t) = A \frac{d}{dt} \frac{\rho \rho_{pi} h(t) x_B(t)}{PM_L(t)} \quad (4)$$

En la ecuación 3, $M(t)$ muestra la masa del líquido más la masa del vapor dentro del tanque, $PM(t)$ denota el peso molecular. De la ecuación 4, A representa el área del tanque, ρ es la densidad, $h(t)$ es la altura del tanque y $PM_L(t)$ es el peso molecular del líquido.

Seguidamente se realizó el balance de energía en estado dinámico del tambor de evaporación obteniendo las ecuaciones que se muestran a continuación:

$$F(t)C_{pi}(t)T_i(t) - L(t)C_{pi}(t)T(t) - V(t) \left(C_{pi}^y(t)T(t) + \lambda_{vap}^y(t) \right) + Q = A \frac{d}{dt} \frac{C_v(t)\rho_{pi}(t)h(t)T(t)}{PM_L(t)} \quad (5)$$

Siendo $C_{pi}(t)$ la capacidad calorífica a presión constante, $T_i(t)$ muestra la temperatura de entrada, $T(t)$ representa la temperatura, $C_{pi}^y(t)$ se refiere a la capacidad calorífica en el benceno, $\lambda_{vap}^y(t)$ denota el calor de evaporación en el benceno, Q es la cantidad de calor transferida, $C_v(t)$ simboliza la capacidad calorífica del vapor.

El calor máximo requerido para vaporizar toda la mezcla, viene dado por la siguiente expresión:

$$Q_{max} = F(\lambda_{benceno}Z_i + \lambda_{tolueno}(1 - Z_i)) \quad (6)$$

Q_{max} es la cantidad de calor máxima, $\lambda_{benceno}$ es el calor de evaporación del benceno, Z_i es la concentración a la entrada, $\lambda_{tolueno}$ es el calor de evaporación del tolueno.

Mediante la ecuación 7 se determina el balance de energía en estado estacionario.

$$\bar{F}\bar{C}_{pi}\bar{T}_i - \bar{L}\bar{C}_{pi}\bar{T} - \bar{V}(\bar{C}_{pi}^y\bar{T} + \bar{\lambda}_{vap}^y) + Q = 0 \quad (7)$$

Luego como se tiene un serpentín para calentar la mezcla dentro del tambor de evaporación, se determinó el balance de energía en el serpentín, generándose como resultado la siguiente ecuación:

$$0 = \lambda_{vap}\bar{W} - UA(\bar{T}_{serp} - \bar{T}) \quad (8)$$

\bar{W} es el flujo de vapor de agua del serpentín, U coeficiente de transferencia de calor, \bar{T}_{serp} es la temperatura del serpentín.

En la Figura 3, se puede observar el sistema en *Simulink*, el cual se ha diseñado en función de las ecuaciones del proceso obtenidas anteriormente.

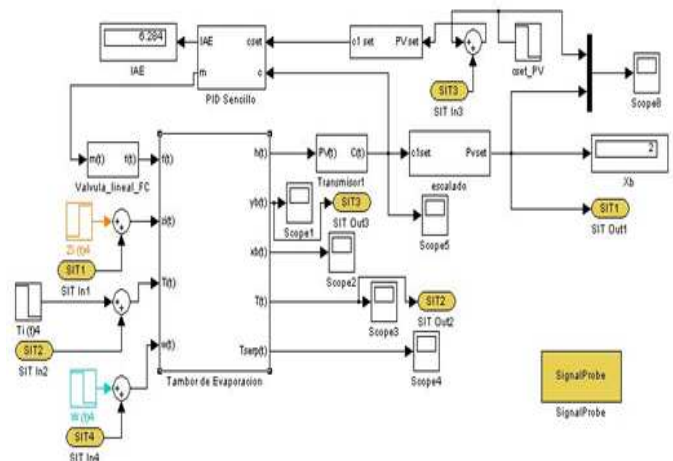


FIG 3. Simulación del proceso en *Simulink*

Después de realizar el análisis de sensibilidad sobre el proceso se determinó que la variable controlada es el nivel de flujo dentro del tambor de evaporación ($f(t)$) y las variables manipuladas son: la concentración del flujo de entrada ($Z_i(t)$), la temperatura del flujo de entrada ($T_i(t)$), y el flujo de vapor que circula por el serpentín ($W(t)$).

En la sintonización del controlador se realizaron diferentes pruebas de métodos de control PID, entre las que están; la sintonización por *Dhalin*, por *Ziegler-Nichols*, y por minimización de IAE (Integral Absoluta del Error) [6]. Se seleccionó el método de sintonización de *Dhalin*, ya que fue el que presentó mejores resultados, los cuales se verificaron determinando el IAE en cada caso.

III. DESARROLLO DEL SISTEMA SCADA

El sistema SCADA en estudio se desarrolló utilizando el software *LabVIEW* provisto por *National Instruments* [7,8,9,10]. El mismo tiene como objetivo primordial, evaluar el desempeño de los nuevos *Toolkit* del software *LabVIEW* para probar las eficiencias de estas herramientas en el desarrollo de sistemas SCADA, así como la comunicación que se establece entre los equipos que conforman.

Un SCADA es un sistema de Supervisión, Control y Adquisición de Datos en tiempo real en un proceso. El desarrollo de sistemas SCADA con *LabVIEW* ha sido implementado por los ingenieros con el fin de conectar los dispositivos de campos a través de OPC (*OLE (Object Linking and Embedding) For Process Control*) [8]. Los sistemas SCADA ofrecen una serie de prestaciones entre las que podemos mencionar las siguientes:

- Posibilidad de crear paneles de alarma.
- Generación de datos históricos.
- Ejecución de programas de control.
- Supervisión remota del proceso.
- Interfaz gráfica del proceso agradable al usuario.
- Almacenamiento de la información.
- Instalación sencilla.

Entre los elementos que componen los sistemas SCADA tenemos el entorno visual o interfaz entre el operador y la máquina, la unidad maestra o central que es la encargada de ejecutar las acciones de control, el almacenamiento de la información y procesamiento de los datos, las unidades remotas que son los elementos que envían la información a la unidad central, el sistema o protocolo de comunicación, y los elementos sensores – transductores [11].

La comunicación en los sistemas SCADA se puede realizar a través de una red, de forma inalámbrica, por satélite, entre otras. Para el desarrollo de este sistema se utilizó una comunicación de red tipo LAN (*Local Area Network*), en la que la información es transmitida por direccionamiento IP (*Internet Protocol*) desde la unidad maestra o servidor principal hasta las unidades clientes utilizando un cable de red [12].

El primer paso para el desarrollo del SCADA fue la creación de la interfaz gráfica del proceso en *LabVIEW* [13], esto se realizó utilizando las imágenes que provee el módulo *DSC (Distributed Systems Control)* [7,14]. En ella se creó la columna de destilación con las variables de interés para el proceso, tales como: El punto de consigna, la temperatura del serpentín, el flujo del serpentín, y la concentración del flujo de entrada. La Figura 4 ilustra el panel de alarmas, el período de muestreo y los histogramas del proceso.

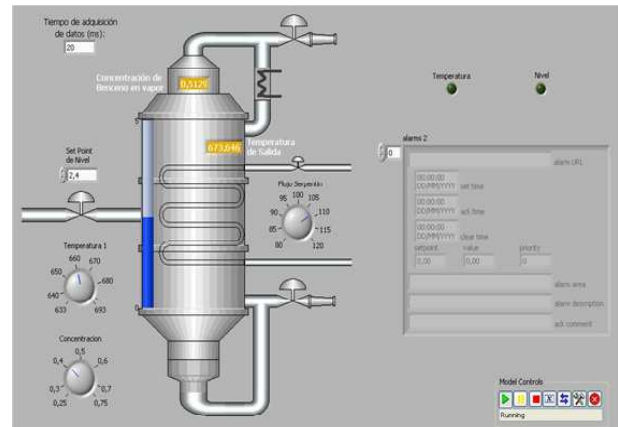


FIG 4. Interfaz gráfica del SCADA

Con el módulo *DSC* se pueden desarrollar cientos de conexiones para compartir variables de supervisión y control de procesos virtuales y reales, reduciendo el tiempo de desarrollo del sistema a implementar.

Una vez creada la interfaz gráfica, se utilizó el módulo *SIT (Simulation Interfaz Toolkit Connection Manager)* de *LabVIEW* [8], con el cual se realizará la comunicación entre la simulación del proceso que se estará ejecutando en *Simulink* y el sistema SCADA que se ejecutará en *LabVIEW*. Con este se pueden seleccionar las variables de intercambio de información entre ambos software y se ajustan los parámetros de simulación para que sean compatibles y no se produzcan errores en la ejecución. Además él permite establecer la comunicación entre ambos software, mediante el protocolo *TCP/IP (Transmission Control Protocol/Internet Protocol)* en la red LAN.

La Figura 5 muestra el diagrama de bloques donde se describe el proceso que se realiza para la intercomunicación entre los software utilizados en el proceso.

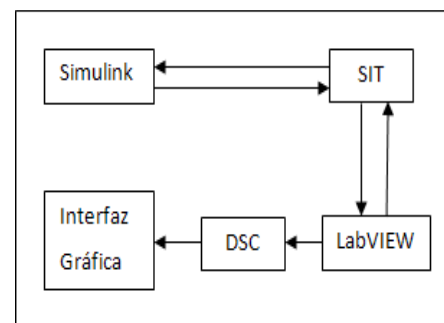


FIG 5. Esquema de la interconexión entre los software

IV. RESULTADOS

El controlador implementado cumple con su función para variaciones en las variables manipuladas de $\pm 20\%$ de su valor en estado estacionario en cada uno de sus parámetros (K_c , T_i , T_d), pudiéndose observar que el sistema presenta pequeños sobre disparos o una ligera disminución en el tiempo de establecimiento [6,12]. El controlador para el proceso, presenta una alta robustez. Todo esto se verificó determinando el IAE. La Figura 3 mostrada anteriormente se presenta el sistema implementado en *Simulink*, el cual es el encargado de simular el proceso del tanque de Benceno-Tolueno.

En la Figura 6 se observa el comportamiento del sistema ante una perturbación, la cual se hace en el servidor, que es la unidad donde se ejecuta el sistema SCADA a través del *LABVIEW*. La repuesta a esa perturbación es obtenida del proceso que es corrido en el programa *Simulink*.

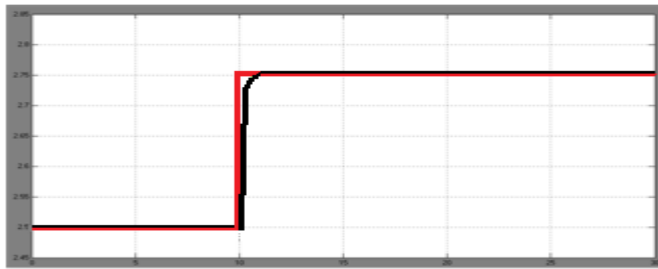


FIG 6. Curva de reacción del nivel del tanque ante una perturbación controlada con sintonización por *Dhalin*. IAE= 0.94

Para probar el funcionamiento del sistema, se estableció una secuencia de supervisión sobre el sistema de control de la torre de destilación, luego se establecieron los parámetros de la comunicación y las respuestas del sistema SCADA, que fue establecido entre los dos programas antes mencionados. Estas son las pruebas a la cuales fue sometido el sistema SCADA y de las cuales se obtuvo el conjunto de respuestas deseadas, según la secuencia de funcionamiento pautada para el control y supervisión. En la Figura 7, se puede observar la activación de la alarma a causa de la perturbación aplicada al sistema.

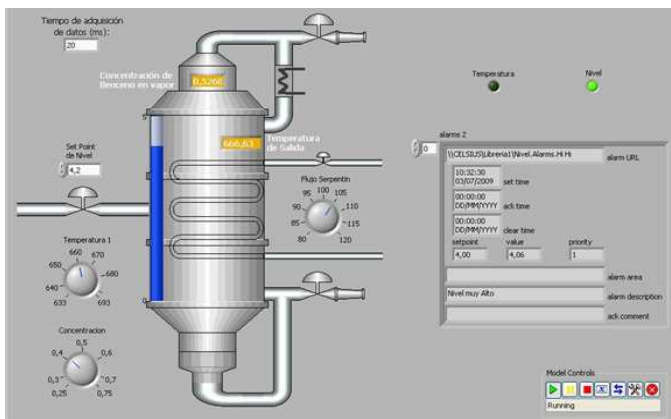


FIG 7. Perturbaciones en el SCADA y activación de alarma por nivel alto

La Figura 8 muestra la gráfica obtenida en *Simulink* para la perturbación realizada en el proceso.



FIG 8. Gráfico de tendencia histórica del SCADA. (a) Nivel (superior), (b) Temperatura (inferior)

La repuesta a las perturbaciones aplicadas al sistema se almacenan en una base de datos, la misma es creada en el servidor, los valores almacenados en ella se obtienen del proceso que está corriendo en el cliente los cuales son transportados por el sistema al servidor para ser almacenados y estudiados, dichos valores se guardan en una tabla en Excel, con lo cual posteriormente se pueden realizar estudios estadísticos complementarios según el interés del administrador del sistema.

En la Tabla 1 se presenta una muestra de las variables de interés obtenidas en el proceso.

TABLA 1.
 BASE DE DATOS DEL SCADA.

Numero de muestras	Nivel del Tambor (m)	Concentración de Benceno (%)	Temperatura de Salida (°F)
25	2,2000	0,5109	664,5407
26	2,5979	0,5157	664,4222
27	2,7965	0,5202	663,9216
28	2,9450	0,5228	664,5346
29	3,1408	0,5244	664,5773
30	3,2005	0,5216	664,3157
31	3,2001	0,5215	664,3170
32	2,9950	0,5186	664,6691
33	2,7999	0,5154	664,8719
34	2,6050	0,5129	664,9302
35	2,3984	0,5123	664,8451
36	2,4005	0,5134	664,4910
37	2,4001	0,5134	664,4907
38	2,4000	0,5134	664,4907
39	2,4000	0,5134	664,4907
40	2,5991	0,5157	664,4378
41	2,8007	0,5182	664,2236
42	3,0000	0,5197	664,3546
43	3,2005	0,5216	664,3144
44	3,4014	0,5233	664,2661
45	3,6014	0,5249	664,2392
46	3,8027	0,5266	664,1827
47	4,0029	0,5280	664,1602
48	4,2025	0,5293	664,1356

Otro de los resultados que se obtuvo fue el intercambio de datos del sistema SCADA en una red tipo LAN, es decir, el

sistema SCADA puede ser manejado con *LabVIEW* desde diferentes computadores utilizando direccionamiento IP para el intercambio de datos en el sistema.

V. CONCLUSIONES

En el desarrollo del sistema SCADA se probó la funcionalidad, flexibilidad y seguridad de las herramientas proporcionadas por *National Instruments* para el diseño de sistemas SCADA, obteniéndose resultados satisfactorios. Esta herramienta es muy ágil y permite la creación e implementación de estos sistemas en forma rápida y eficiente.

El protocolo de comunicación utilizado para la supervisión remota del sistema SCADA presentó resultados exitosos, los tiempos de respuesta del sistema fueron bastante rápidos cuando se realizaron las pruebas de desempeño del mismo.

Las herramientas desarrolladas por *National Instruments* para el desarrollo de sistemas de supervisión es eficaz, ya que los *Toolkits* que provee mejoran su desenvolvimiento, permitiendo establecer un protocolo de comunicación directa con otras herramientas de software como lo es MATLAB, para la elaboración del modelado matemático del proceso que se desea implementar.

El tipo de control que ejercen los sistemas SCADA sobre un proceso industrial es de tipo supervisório, y estos sistemas permiten hacer variaciones en los parámetros de control más importante o críticos que tiene el proceso que se está implementando de forma remota, es decir, a una distancia segura para los operadores.

VI. REFERENCIAS BIBLIOGRÁFICAS

[1] Aquilino Rodriguez (2007). “**Sistemas SCADA**”. 2da Ed. Editorial Alfaomega. México. México.

- [2] Mendiburo D. Henry (2005). “**Sistemas SCADA**”. Consultado en Enero 2010, Disponible en: <http://www.galeon.com/homd/pdf/scada.pdf>
- [3] Himmelblau David (1997). “**Principios Básicos y Cálculos en Ingeniería Química**”. 6ta Ed. PRENTICE-HALL HISPANOAMERICANA. Naucalpan de Juárez, México.
- [4] Roncancio Henry Antonio (2001). “**Tutorial de LabVIEW**” Universidad Distrital “Francisco José de Caldas”. Consultado en Marzo 2009, Disponible en: <http://perso.wanadoo.es/jovilve/tutoriales/016tutorlabview.pdf>
- [5] Yong Xiong; Bin Qin; Minjun Wu; Jun Yang y Mingwu Fan Huazhong (2007). “**LabVIEW and MatLAB Based Virtual Control System for Virtual Prototyping of Cyclotron**”. , University of Science and Technology, Wuhan, P. R. China, IEEE Proceeding Instrumentation, Controls
- [6] Smith Carlos y Corripio Armando (1991). “**Control Automático de Procesos. Teoría y Práctica**”. 1era Ed. Editorial Limusa. México DF. México.
- [7] National Instrument (2001). “**Database Connectivity Toolset User Manual**” Consultado en Marzo 2009, Disponible en: www.ni.com/pdf/manuals/321525c.pdf
- [8] National Instruments (2003) “**LabVIEW Simulation Interface Toolkit User Guide, National Instruments**” Consultado en Marzo 2009, Disponible en: <http://ftp.ni.com/pub/devzone/tut/manual.pdf>
- [9] National Instrument (2009). “**ActiveX and LabVIEW**”. Consultado en Marzo 2009, Disponible en: <http://zone.ni.com/devzone/cda/tut/p/id/2983>
- [10] National Instrument (2006). “**Working with .m File Scripts in NI LabVIEW for Text Based Signal Processing, Analysis, and Math**”. Consultado Mayo 2009. Disponible: <http://zone.ni.com/devzone/cda/tut/p/id/4854>
- [11] Piedrafita Moreno, Ramón (2004) “**Ingeniería de la Automatización Industrial**”. 2da Ed. Editorial Alfaomega.
- [12] Creus S. Antonio (1997). “**Instrumentación Industrial**”. 6ta Ed. Marcombo Boixareu Editores. Barcelona España.
- [13] González, Rodrigo y Pradines Rodrigo. (2007) “**Análisis de Software para Desarrollo de Entorno Gráfico LabVIEW y Propuesta de Implementación para Laboratorio en el Instituto de Electricidad y Electrónica en Universidad Austral de Chile**”. Valdivia, Chile.
- [14] Kasemir, K.U; Pieck, K.U y. Dalesio, L.R (2001). “**Integrating Labview Into a Distributed Computing Environment**”. , San Jose, California(USA) Consultado Abril 2009. Disponible: <http://arxiv.org/pdf/cs/0111001>.

Diseño de un Sistema de Mantenimiento basado en Agentes Inteligentes

Flor Narciso, Addison Ríos-Bolívar, Isabel Besembel

Resumen— En este artículo se presenta el diseño de un sistema multiagentes (SMA) de mantenimiento, encargado de gestionar el mantenimiento correctivo, preventivo y predictivo de una planta industrial. Este SMA consta de los agentes Manejador de Fallas, Planificador, Ejecutor y Supervisor de Confiabilidad. Para el diseño se utiliza la Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes que permite desarrollar SMA en ambientes de automatización y control de procesos. La fase de diseño de esta metodología involucra el modelado del SMA que genera como productos los diagramas de clases y de componentes, la tabla de definición del universo de clases y tipos de datos abstractos (TDAs) de cada agente, la tabla de definición formal de la clase que representa a cada agente, las tablas de especificación formal de los métodos de cada clase y el modelado de la plataforma del SMA representado mediante un diagrama de despliegue.

Palabras claves—Automatización Industrial, Mantenimiento, Modelado de Sistemas Multiagentes, Sistemas Multiagentes.

I. INTRODUCCIÓN

En la actualidad, los sistemas de automatización industrial pueden ser modelados utilizando estructuras jerárquicas que integran aspectos operacionales (nivel de dispositivos de campo), de planificación, de supervisión y control local (nivel táctico) y de gerencia de los procesos (nivel estratégico) [1]. Esto permite concederle un enfoque distribuido a las actividades realizadas por los diferentes componentes de estos sistemas, cuya complejidad, desde los puntos de vista funcional y de diseño e ingeniería, conjuntamente con su naturaleza distribuida, hacen de los SMA una herramienta de *software* apropiada para ser utilizada en la automatización y control de procesos industriales [1], [2-7], debido a ciertas características presentes en los agentes que los conforman, tales como autonomía, comunicación, sociabilidad, reactividad, inteligencia, adaptabilidad, movilidad, cooperación, colaboración, competencia y proactividad [8-11].

Entre las actividades propias de la automatización de un proceso industrial o productivo se encuentran el control de

procesos, el mantenimiento, el manejo de situaciones anormales o fallas, entre otras, de las cuales se tiene que el mantenimiento constituye uno de los factores más importantes para garantizar la efectividad y eficiencia de los procesos productivos, la disminución de los costos, así como también la disponibilidad de la función de los equipos e instalaciones con la finalidad de atender, adecuadamente y a tiempo, el proceso productivo ó de servicios, con confiabilidad, seguridad, preservación del medio ambiente y costos adecuados aún bajo condiciones inciertas ó de contingencias operacionales [12], [13].

Las formas o tipos de mantenimiento han evolucionado a través del tiempo debido principalmente a la complejidad de los procesos industriales y a la necesidad de optimizar los costos, mejorando sustancialmente el funcionamiento con el objeto de garantizar la alta productividad de una empresa.

Las empresas con alto potencial de desarrollo apuntan hoy en día hacia la categorización de sus prácticas de mantenimiento como Mantenimiento Clase Mundial [14]. Sin embargo, a partir del surgimiento del paradigma de Mantenimiento Centrado en Confiabilidad (MCC), los paradigmas de mantenimiento que se han desarrollado hasta el presente, preservan sus objetivos en base a los índices de confiabilidad esperados para cada uno de los entes involucrados en el mantenimiento. Por tanto, la Confiabilidad es el punto central del mantenimiento, que ha de ser optimizado haciendo uso de las distintas metodologías existentes, según sean los propósitos evolutivos del mantenimiento dentro de la empresa. Las tendencias se dirigen hacia la integración de todos los activos de la empresa, refiriéndose entonces a la Gestión de Activos, la cual implica un mantenimiento optimizado gracias al uso de todos los recursos materiales, humanos y organizacionales con miras hacia una alta confiabilidad respecto a todos los activos de la empresa.

El mantenimiento está entonces en reacción continua ante nuevas expectativas, entre ellas una mayor importancia a los aspectos de seguridad y del medio ambiente, un conocimiento creciente de la relación mantenimiento-calidad, y requerimientos cada vez mayores de mejorar la disponibilidad ante el control de los costos de mantenimiento.

Es importante resaltar, nuevamente, que en la actualidad, el objetivo de los paradigmas del mantenimiento es lograr evitar todas las fallas no previstas, para lo cual se requieren de la construcción, diseño e implantación de sofisticados sistemas de mantenimiento que permitan dotar de inteligencia a las

Artículo recibido el 15 de Noviembre de 2009. Este artículo fue financiado por el FONACIT bajo el proyecto No. 2005000170.

F.N., A.R.B. e I.B. están con la Universidad de Los Andes, Núcleo La Hechicera, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Mérida, Estado Mérida, Venezuela, Tlf. +58-274-2402981, Fax: +58-274-2402979, E-mail: fnarciso@ula.ve, ilich@ula.ve, ibc@ula.ve.

funcionalidades (manejo de fallas, planificación, ejecución, supervisión, evaluación de confiabilidad, etc.), que dentro del contexto operacional son requeridas para minimizar los efectos de las contingencias o fallas presentes en los procesos productivos.

En este artículo se presenta el diseño del SMA de Mantenimiento propuesto en [15] como una alternativa para realizar el mantenimiento correctivo, preventivo y predictivo de procesos productivos en ambientes de automatización y control de procesos. Este SMA que está conformado por los agentes Manejador de Fallas, Planificador, Ejecutor y Supervisor de Confiabilidad, se encarga de realizar actividades que permiten preservar y/o restablecer las condiciones operacionales de una planta industrial, a fin de garantizar la seguridad del personal, del ambiente y de las instalaciones, así como el funcionamiento apropiado de los equipos y la calidad y cantidad de producción. Cabe destacar que la inteligencia de los agentes del SMA de Mantenimiento se ha considerado en el contexto de responder adecuadamente ante situaciones operacionales inapropiadas ó bajo contingencia.

Para el diseño del SMA de Mantenimiento se utiliza la Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes [7]. Previo a la fase de diseño han sido desarrolladas las fases de conceptualización y análisis de la metodología [7], [16-18]. La fase de conceptualización genera como productos la descripción inicial del SMA en lo referente a los elementos que lo conforman, los procesos que realizan estos elementos y las relaciones existentes entre ellos. La fase de análisis permite obtener la especificación detallada de los elementos identificados en la fase de conceptualización, en términos de los modelos de agente, tareas, inteligencia, coordinación y comunicación.

La fase de diseño permite describir detalladamente la estructura, los componentes de la estructura y la plataforma del SMA. Como resultado de esta fase se obtiene un diseño que puede ser implementado y probado siguiendo los pasos establecidos en la fase de codificación y pruebas de la metodología, la cual genera como producto final un sistema de ingeniería orientado a agentes.

II. SMA DE MANTENIMIENTO

El SMA de Mantenimiento se compone de cuatro agentes autónomos que interactúan entre sí y colaboran entre ellos para alcanzar los objetivos de diseño. Las diversas funcionalidades y objetivos de este SMA se han repartido entre estos agentes, tal y como lo establece la definición de SMA, por lo que se tiene que:

- El agente *Manejador de Fallas* se encarga de detectar fallas técnicas en el proceso de producción en términos del monitoreo de las variables controladas y del proceso de mantenimiento, a partir de los datos asociados al proceso controlado, y a su vez, está en capacidad de realizar, de ser necesario, el diagnóstico de fallas.
- El agente *Planificador* se encarga de generar los planes de mantenimiento correctivo y el mantenimiento

predictivo solicitado por el agente Manejador de Fallas y el agente Supervisor de Confiabilidad, respectivamente, además, planifica el lapso de tiempo en el cual se deben realizar las tareas de planificación para el mantenimiento preventivo y re-planificación en caso de mantenimiento predictivo.

- El agente *Ejecutor* se encarga de poner en ejecución las tareas planificadas para el mantenimiento. En complemento de sus actividades, este agente imparte la información resultante al resto del SMA.
- El agente *Supervisor de Confiabilidad* se encarga de supervisar la confiabilidad del proceso de producción en términos del monitoreo de las variables controladas y del proceso de mantenimiento, a partir de los datos asociados al proceso controlado. Asimismo puede solicitar, si es necesario, el diagnóstico y la detección de fallas y/o cambios en los planes de mantenimiento, constituyendo una herramienta práctica que facilita las actividades llevadas a cabo por ingenieros de optimización y que permiten garantizar la continuidad operativa de una planta industrial.

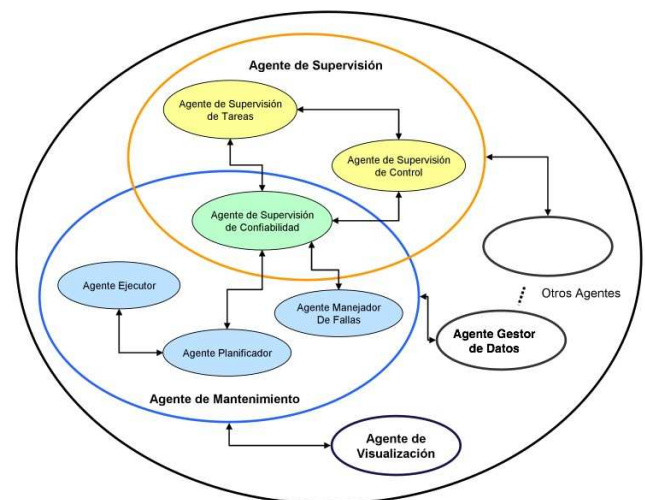


FIG. 1. Modelo de referencia del SMA de Mantenimiento.

En la Fig. 1 se muestra el modelo de referencia del SMA de Mantenimiento, el cual permite gestionar el mantenimiento correctivo, preventivo y predictivo a partir de los servicios prestados por cada uno de los cuatro agentes que lo conforman, tales como: monitorear procesos, monitorear mantenimiento, analizar confiabilidad, manejar fallas, revisar equipos, generar plan de emergencia, re-planificar mantenimiento y ejecutar planes y tareas de mantenimiento, lo que lo convierte en una herramienta apropiada para la toma de decisiones y una opción a considerar cuando se requiere automatizar procesos industriales. También puede observarse en esta figura que el SMA de Mantenimiento es a su vez un subsistema de un SMA de Automatización Industrial para Producción y que sus agentes pueden comunicarse con agentes de este otro SMA, como por ejemplo con el agente Gestor de Datos [19-21], el cual, entre otras cosas, le indica las

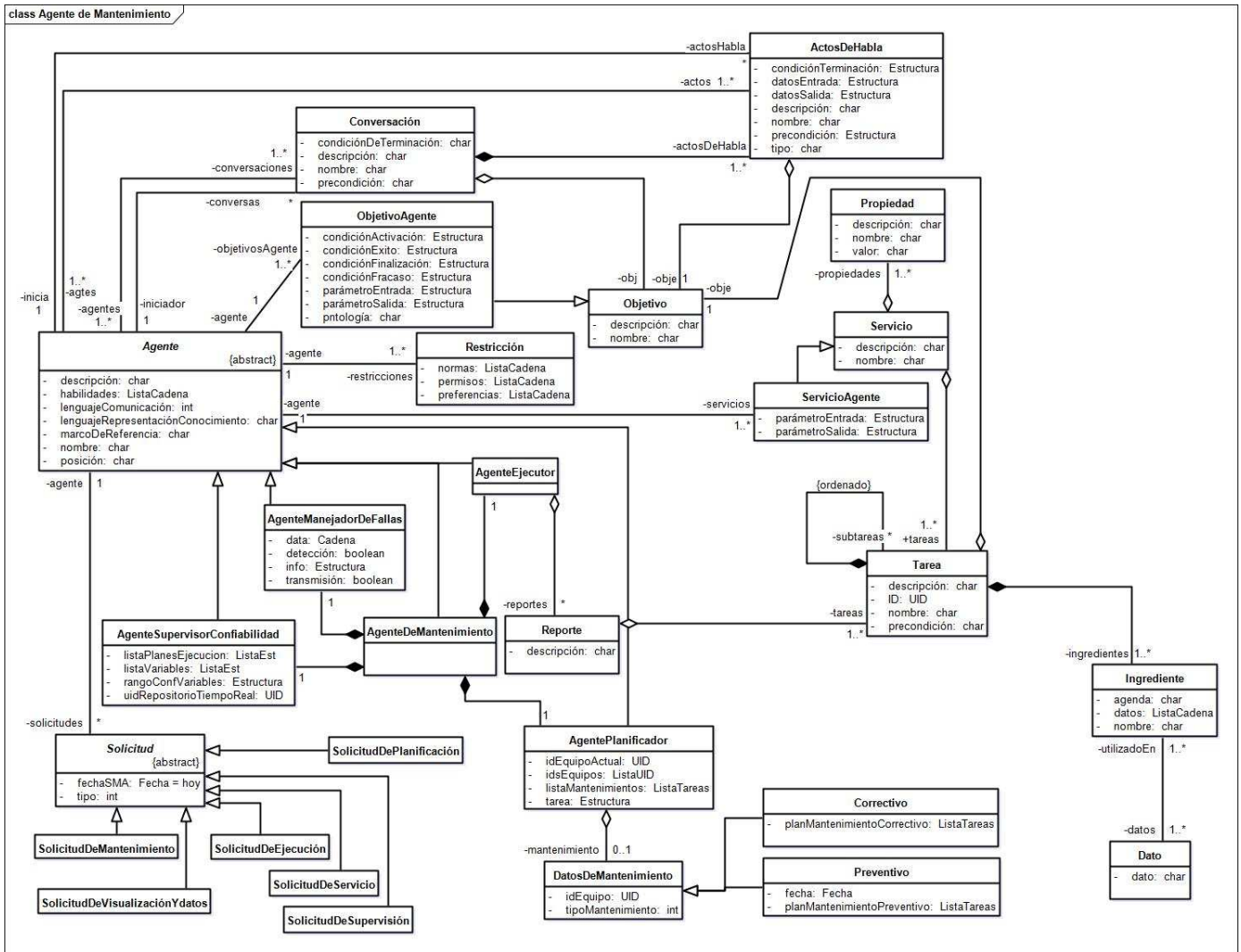


FIG. 2 Diagrama de Clases UML.

características de cada uno de los dispositivos de planta y le permite llevar un historial de las fallas, o con el agente de Visualización [7], [21], [23] cuando requiera desplegar en algún dispositivo de salida, la(s) alarma(s) correspondiente(s) a una falla detectada o los resultados de un servicio, monitorear un proceso o una tarea de mantenimiento, desplegar la información obtenida mediante la ejecución de una tarea realizada por un agente, entre otras.

III. DISEÑO DEL SMA DE MANTENIMIENTO

En esta fase se elabora el modelo de diseño detallado del SMA a partir de los modelos generados en la fase de análisis y se generan como productos los diagramas de clases y de componentes, la tabla de definición del universo de clases y tipos de datos abstractos (TDAs) de cada agente, la tabla de definición formal de la clase que representa a cada agente y las tablas de especificación formal de los métodos de cada clase. Además, en esta fase se realiza el modelado de la plataforma del SMA que se representa mediante un diagrama de despliegue.

A. Modelo de diseño detallado del SMA de Mantenimiento

Para elaborar el modelo de diseño de un SMA, la Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes propone el uso del paradigma de orientación por objetos, haciendo uso de algunos de los diagramas propios del Lenguaje de Modelado Unificado (UML) y la Técnica de Desarrollo de Sistemas de Objetos (TDSO), la cual es una herramienta para la especificación formal de modelos orientados a objetos [7].

En la Fig. 2 se presenta el diagrama de clases de UML que describe la estructura del SMA de Mantenimiento (modelado estructural), en el cual se muestran las clases descritas en términos de sus atributos, así como las relaciones entre ellas.

El diagrama de componentes, que se muestra en la Fig. 3, representa la arquitectura de software del SMA de Mantenimiento por componentes (elementos de software). En este diagrama se muestran los diferentes componentes, junto con sus interfaces para el soporte de la comunicación entre los mismos. El componente *ManejaAgentes* contiene las clases

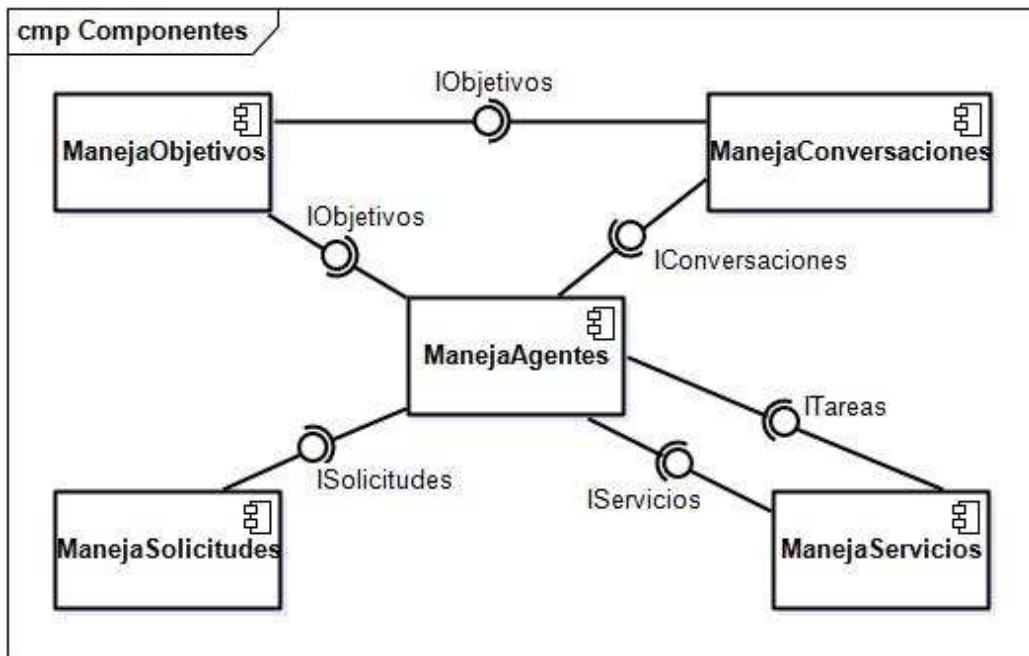


FIG. 3 Diagrama de Componentes.

que soportan los diferentes agentes incluida la clase abstracta *Agente* y las clases *Restricción*, *DatosDeMantenimiento*, *Correctivo*, *Preventivo* y *Reporte*. El componente *ManejaSolicitudes* contiene las clases con prefijo *Solicitud*. El componente *ManejaObjetivos* contiene las clases *Objetivo* y *ObjetivoAgente*. El componente *ManejaConversaciones* incluye las clases *Conversación* y *ActosDeHabla*. Por último, el componente *ManejaServicios* contiene el resto de las clases que conforman el diagrama de clases presentado en la Fig. 2.

Para cada uno de los agentes del SMA de Mantenimiento se realiza la definición del universo de clases y tipos de datos abstractos (TDAs) utilizando la tabla de TDSO destinada para tan fin. Estas tablas pueden consultar en [15], a modo de ejemplo, en la Tabla 1 se presenta la definición del universo de clases y TDAs del agente Ejecutor.

TABLA 1
 UNIVERSO DE CLASES Y TDAs DEL AGENTE EJECUTOR.

09/04/08		Versión 1.0	
Universo de clases y TDAs AgenteEjecutor			
{Colección de clases y TDAs requerida para implantar el Agente Ejecutor}			
1	AgenteEjecutor	<i>AgenteEjecutor</i> : Instancia que se encarga de realizar tareas de mantenimiento y generar reportes.	
2	Reporte	<i>Reporte</i> : TDA que especifica el resultado de la realización de una tarea de mantenimiento.	
3	Tarea	<i>Tarea</i> : TDA que especifica una tarea de mantenimiento.	
4	Logico	<i>Logico</i> : Tipo lógico conformado por los valores cierto y falso.	

Tal y como se muestra en [15], a cada agente le corresponde una clase definida en términos de sus atributos y métodos, por tanto para cada una de ellas se realiza su definición formal. En la Tabla 2 se muestra la definición formal de la clase AgenteEjecutor.

TABLA 2
 DEFINICIÓN FORMAL DE LA CLASE AGENTE EJECUTOR.

20/11/08	1 AgenteEjecutor ()		Versión 1.0
{Ejecuta una tarea de mantenimiento y genera los reportes}			
Especificación de atributos:			
	reporte: Reporte	<i>reporte</i> : Atributo de la clase Reporte que especifica el resultado de la ejecución de la tarea de mantenimiento.	
Especificación sintáctica:			
1	agenteEjecutor()	<i>agenteEjecutor()</i> : Constructor de una instancia del AgenteEjecutor.	
2	realizarTareaMantenimiento(tarea: Tarea): Logico	<i>realizarTareaMantenimiento()</i> : Método que se encarga de ejecutar una tarea de mantenimiento	
3	recibirTareaMantenimiento(Tarea: Tarea): Logico	<i>recibirTareaMantenimiento()</i> : Método que se encarga de recibir una Tarea de Mantenimiento.	
4	enviarReporte(Reporte: Reporte): Logico	<i>enviarReporte()</i> : Método que se encarga de enviar el reporte generado de la ejecución de una tarea de mantenimiento.	
5	destruirAgenteEjecutor()	<i>destruirAgenteEjecutor()</i> : Destructor del TDA AgenteEjecutor.	

A modo de ejemplo, en la Tabla 3 se presenta la especificación formal del método *realizarTareaMantenimiento* perteneciente a la clase *AgenteEjecutor*. La especificación formal de todos los métodos definidos para cada uno de los agentes del SMA de Mantenimiento puede ser consultada en [15]. Dichos métodos definen las características funcionales o comportamiento de cada uno de los agentes del SMA de Mantenimiento.

B. Modelo de diseño de la plataforma del SMA de Mantenimiento

El modelado de la plataforma del SMA de Mantenimiento

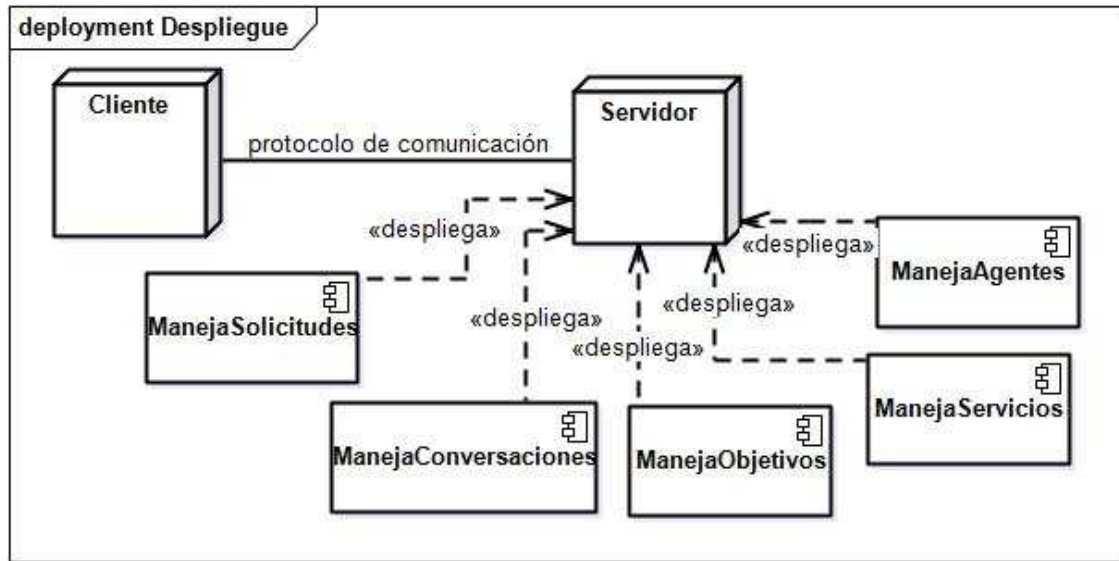


FIG. 4 Diagrama de Despliegue.

se representa mediante el diagrama de despliegue que se muestra en la Fig. 4 (modelado arquitectónico de hardware), en el cual presentan los componentes mostrados en la Fig. 3 desplegados en el nodo servidor del diagrama.

TABLA 3
 UNIVERSO DE CLASES Y TDAS DEL AGENTE EJECUTOR.

09/04/08		Versión 1.0	
1,2 (Transformador, Público)			
realizarTareaMantenimiento(tarea: Tarea): Logico			
{Realiza una tarea de mantenimiento}			
{pre: Haber recibido la tarea de mantenimiento}		{pos: Tarea ejecutada}	
1	Si (enviarSolicitudVisualizacion(solicitud)) tareaRealizada = CIERTO SiNo tareaRealizada = FALSO FinSi	enviarSolicitudVisualizacion(): Envía al agente de Visualización una solicitud de visualización. solicitud: Atributo de la clase Tarea que contiene la solicitud a enviar. tareaRealizada: Variable de tipo Logico.	
2	devuelve(TareaRealizada)	devuelve(): Retorna un valor lógico, CIERTO si se realizó la tarea, FALSO si no.	

IV. CONCLUSIONES

Con el diseño del SMA de Mantenimiento presentado en este trabajo queda demostrado que es posible desarrollar una aplicación de software basada en agentes inteligentes, concebida como un SMA, para el mantenimiento de sistemas de producción industrial.

El SMA de Mantenimiento está dotado de funcionalidades para el manejo de fallas, planificación, ejecución y supervisión de confiabilidad, lo que permite atender las tareas propias del

mantenimiento preventivo y correctivo y, a su vez, responder apropiadamente ante situaciones de funcionamiento bajo falla o contingencia, dándole un carácter de mantenimiento predictivo, tarea fundamental en la concepción de Sistemas de Mantenimiento Centrados en Confiabilidad. Asimismo, en las fases previas al diseño y en el diseño propiamente dicho de los agentes Supervisor de Confiabilidad y Manejador de Fallas se han considerado todos aquellos aspectos que garantizan la seguridad del personal, del ambiente y de las instalaciones, así como el funcionamiento apropiado de los equipos y la calidad y cantidad de producción.

El diseño de SMA de Mantenimiento ha sido desarrollado utilizando la Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes, generándose los diagramas de clases y de componentes, la tabla de definición del universo de clases y tipos de datos abstractos de cada agente, la tabla de definición formal de la clase que representa a cada agente, las tablas de especificación formal de los métodos de cada clase y el modelado de la plataforma del SMA representado mediante un diagrama de despliegue, lo que permite una fácil implementación, sin mayores requerimientos, a partir de plataformas de software típicas.

El diseño del SMA de Mantenimiento desarrollado permite que su implementación pueda ser concebida dentro de un marco referencial para la automatización integrada de procesos productivos, lo que facilita su integración con diferentes sistemas de gestión estratégica, táctica y operacional de la planta industrial.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el FONACIT bajo el proyecto No. 2005000170, institución a la cual los autores expresan sus agradecimientos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Ríos-Bolívar, M. Cerrada, J. Durán and F. Narciso. **“Implantando Sistemas de Control usando Agentes Inteligentes”**. Memorias del VI Congreso Nacional de Ingeniería Mecánica (CONIM'2006), Mérida, Venezuela, 2006.
- [2] J. Aguilar, M. Cerrada, F. Hidrobo, G. Mousalli and F. Rivas. **“Modelado de Referencia de un Sistema de Control Distribuido Inteligente utilizando Sistemas Multiagentes”**. Memorias del II Coloquio de Automatización y Control (CAC'2000), Puerto La Cruz, Venezuela, 2000.
- [3] J. Aguilar, M. Cerrada, G. Mousalli, F. Rivas, and F. Hidrobo. **“A Multiagent Model for Intelligent Distributed Control Systems”**. Lecture Notes in Artificial Intelligence, Vol. 3681, pp. 191-197, 2005.
- [4] M. Cerrada, J. Cardillo, J. Aguilar and R. Fainete. **“Agents-based Design for Fault Management Systems in Industrial Processes”**. Computers in Industry, Vol. 58, No. 4, pp. 313-328, 2007.
- [5] N. Jennings and S. Bussmann. **“Agent-based Control Systems”**. IEEE Control Systems Magazine, pp. 61-73, 2003.
- [6] V. Marik and D. McFarlane. **“Industrial Adoption of Agent-based Technologies”**. IEEE Intelligent Systems, pp. 27-35, 2005.
- [7] J. Aguilar, I. Besembel, M. Cerrada, F. Hidrobo, and F. Narciso. **“Una Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes”**. Revista Iberoamericana de Inteligencia Artificial, Vol. 12, No. 38, pp. 39-60, 2008.
- [8] S. Franklin and A. Graesser. **“Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents”**. Proceedings of the Third. International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, 1996.
- [9] B. Hayes-Roth. **“An Architecture for Adaptive Intelligent Systems”**. Artificial Intelligence: Special Issue on Agents and Interactivity, Vol. 72, pp. 329-365, 1995.
- [10] P. Maes. **“Modeling Adaptive Autonomous Agents”**. Artificial Life, Vol. 1, pp. 135-162, 1994.
- [11] M. Wooldridge and N.R. Jennings. **“Intelligent Agents: Theory and Practice”**. Knowledge Engineering Review, Vol. 10, No. 2, pp.115-152, 1995.
- [12] J. Aguilar, J. Calderón, V. Bravo, O. Cárdenas, J. González, D. Hernández, L. León, N. Pérez, D. Pernía, A. Ríos, F. Rivas, and O. Terán. **“Levantamiento de Requerimientos y Determinación de los Paradigmas de la Plataforma de Automatización”**. AIT-PDVSA, Mérida, Venezuela, Reporte Técnico, 2004.
- [13] J. Aguilar, H. Arciniegas, A. Bianchi, J. Calderón V. Bravo, O. Cárdenas, J. González, D. Hernández, L. León, N. Pérez, D. Pernía, A. Ríos, F. Rivas, and O. Terán. **“Conceptualización de la Arquitectura de Automatización Industrial para Producción”**. Mérida, Venezuela, AIT- PDVSA, Reporte Técnico, 2003.
- [14] Kardec, Alan. **“Mantenimiento: Función Estratégica”**. Qualitymark, Río de Janeiro, 2002.
- [15] M. Avila, T. Bastidas, M. Bislick, F. Narciso, D. Páez, R. Parra, M. A. Ramírez, C. Rodríguez, and G. Valero. **“Conceptualización, Análisis y Diseño del Agente de Mantenimiento”**. Universidad de Los Andes, Mérida, Venezuela, Reporte Técnico, Abril, 2008.
- [16] F. Narciso, A. Ríos-Bolívar, M. Avila and D. Páez. **“Diseño de un Sistema Multiagentes de Mantenimiento: Agente Manejador de Fallas”**. Memorias del XIII Congreso Latinoamericano de Control Automático/V Congreso Venezolano de Automatización y Control, Mérida, Venezuela, 2008, pp. 982 - 989.
- [17] F. Narciso, A. Ríos-Bolívar, C. Rodríguez and G. Valero. **“Diseño de un Sistema Multiagentes de Mantenimiento: Agente Planificador”**. Memorias del XIII Congreso Latinoamericano de Control Automático/V Congreso Venezolano de Automatización y Control, Mérida, Venezuela, 2008, pp. 967 - 974.
- [18] F. E. Narciso, A. Y. Ríos-Bolívar, M. A. Bislick and R. E. Parra. **“Sistema Multiagentes de Mantenimiento: Diseño del Agente Supervisor de Confiabilidad”**. Memorias del 9º Congreso Interamericano de Computación Aplicada a la Industria de Procesos (CAIP'2009), Montevideo, Uruguay, 2009.
- [19] J. Aguilar, L. León, A. Ríos-Bolívar, M. Cerrada, F. Hidrobo, V. Bravo, F. Rivas, J. Valera, G. Mousalli, J. Calderón, F. Narciso, and W. Rodríguez. **“Descripción Funcional del Medio de Gestión de Servicios, Modelo Arquitectónico del Sistema Multiagente, con la Descripción de los Actores y Casos de Uso de los Mismos”**. Fundacite-ULA, Mérida, Venezuela, Reporte Técnico, Junio, 2005.
- [20] J. Aguilar, L. León, A. Ríos-Bolívar, M. Cerrada, F. Hidrobo, G. Mousalli, F. Narciso, D. Hernández, and I. Besembel. **“Documento de Especificación del Medio de Gestión de Servicios”**. Fundacite-ULA, Mérida, Venezuela, Reporte Técnico, Septiembre, 2005.
- [21] J. Aguilar, L. León, A. Ríos-Bolívar, M. Cerrada, F. Hidrobo, G. Mousalli, F. Narciso, D. Hernández, and I. Besembel. **“Documento de Ingeniería de Implantación del Medio de Gestión de Servicios”**. Fundacite-ULA, Mérida, Venezuela, Reporte Técnico, Octubre, 2005.
- [22] J. Aguilar, L. León, A. Ríos-Bolívar, M. Cerrada, F. Hidrobo, F. Narciso, D. Hernández, and I. Besembel. **“Documento de Especificación del Nivel Superior”**. Fundacite-ULA, Mérida, Venezuela, Reporte Técnico, Noviembre, 2005.