

## PLATAFORMA DE ENSAYO REMOTO DE CONTROLADORES BASADOS EN AUTÓMATAS FINITOS

Daniel Gasa Fonollosa\* Ignacio Garrido Girón\*  
Ramon Costa Castelló\* Luis Basañez Villaluenga\*

\* *Institut d'Organització i Control de Sistemes Industrials  
(IOC)*  
*Escola Tècnica Superior d'Enginyeria Industrial de  
Barcelona (ETSEIB)*  
*Universitat Politècnica de Catalunya (UPC)*  
*Av. Diagonal 647,11*  
*08028-Barcelona*

Resumen: En este artículo se describe uno de los componentes de la plataforma desarrollada en el IOC y destinada a la realización de prácticas experimentales de Automática de forma remota. En concreto, se describe el entorno desarrollado en el área de programación de controladores lógicos programables (PLCs) mediante autómatas finitos. Dicho entorno permite programar el comportamiento de una planta didáctica (ascensor) mediante una descripción de basada en autómatas finitos de tipo *Moore*, y visualizar su evolución mediante realimentación visual y una monitorización remota. El artículo presenta los componentes y la arquitectura de la plataforma así como la aplicación cliente construida. *Copyright © 2005 CEA-IFAC.*

Palabras Clave: Educación en automática, prácticas experimentales, laboratorios remotos, autómatas programables.

### 1. INTRODUCCIÓN

Dentro de la enseñanza de la Ingeniería, y en particular de la Automática, la realización de actividades prácticas experimentales que ilustren las exposiciones teóricas es de vital importancia y difícilmente sustituible por otro tipo de actividades tales como la simulación o las clases magistrales (Sánchez *et al.*, 2001) (Giralt *et al.*, 2002).

Sin embargo, existen diferentes factores que dificultan la realización de dichas actividades prácticas. El gran número de estudiantes y la reducida capacidad de los laboratorios docentes, ya sea en medios o en espacio, hace necesaria una elevada

ocupación de dichos laboratorios. Como contrapartida, el uso intensivo de los laboratorios supone un coste económico importante y permanente debido a la necesidad de personal de atención y vigilancia lo que a veces resulta difícil asumir. A fin de obviar esta dificultad se ha extendido el interés por poner en marcha plataformas que permita la realización de prácticas remotas con un contenido pedagógico similar al de las realizadas de forma presencial.

Las prácticas remotas facilitan a los estudiantes acceso a los laboratorios en horarios sin restricciones y un mayor número de veces. Por otro lado, el acceso remoto permite un elevado grado de ocupación cosa que permite rentabilizar

el coste del equipamiento. Como valor adicional, la manipulación remota evita el mal uso de los equipos, lo que puede alargar notablemente la vida útil de los mismos.

La posibilidad de realizar prácticas remotas es de particular importancia en los cursos semipresenciales ya que introduce una flexibilidad imprescindible en este tipo de estudios. Debido a todo ello, en la actualidad se están implantando numerosos laboratorios remotos para la realización de prácticas de Automática, y, en particular, de Robótica (McKee, 2002) (Torres *et al.*, 2002), aunque existen también otros ámbitos de aplicación como el diseño de controladores (Sánchez *et al.*, 2001) (Puerto *et al.*, 2002) (Vallés *et al.*, 2002) (Giralt *et al.*, 2002).

La realización de prácticas de forma remota requiere que el estudiante pueda percibir sensaciones análogas a las que tendría frente al proceso físico (Sánchez *et al.*, 2001). En el ámbito de la realización de prácticas remotas, esta realimentación sensorial se suele centrar principalmente en la visión. En la actualidad existen diferentes tecnologías para la transmisión en directo de lo que acontece en el entorno de las prácticas, las más utilizadas son el uso de programas específicos (ej. applets JAVA) (McKee, 2002) o el uso de componentes multimedia de conexión a Internet (ej. Servidores Axis) (Guzmán *et al.*, 2002) (Torres *et al.*, 2002). Aunque esta transmisión proporciona unas sensaciones parecidas a la vividas *in situ*, requiere conexiones de altas prestaciones.

En el caso de servidores multimedia, éstos suelen funcionar en paralelo con el experimento por lo que varios usuarios pueden observar simultáneamente lo que acontece en el laboratorio, independientemente de que estén interactuando con él o no. Ello implica la necesidad de indicar a cada estudiante cuándo se ejecuta el experimento que él ha diseñado. Otro tipo de tecnología consiste en grabar en vídeo, en el servidor, la evolución de la planta durante la realización del experimento y posteriormente transmitir la secuencia de vídeo al cliente (Sebastián *et al.*, 2002). Este enfoque introduce un retardo importante entre la realización del experimento y la observación del mismo por parte del estudiante, pero en contrapartida permite observar el experimento con buena calidad independientemente del tipo de conexión, además de poder visualizarlo tantas veces como sea necesario.

En numerosas ocasiones las sensaciones remotas son generadas a partir de entornos simulados (Serrano *et al.*, 1999) (Torres *et al.*, 2002) (Maza *et al.*, 2002). También es usual la combinación de imágenes en directo con simulación (Torres *et al.*, 2002). Aunque este tipo

de procedimientos dista de ser óptimo desde el punto de vista sensorial, permite ofrecer una cierta realimentación en entornos con comunicaciones de bajas prestaciones, a la espera de que la introducción de nuevas tecnologías como las recogidas en el proyecto Internet2, permita utilizar directamente la transmisión de vídeo.

Las técnicas de realidad aumentada (Azuma, 1997), ofrecen un compromiso interesante. En ellas se combina la información sensorial obtenida directamente del entorno y la generada a partir de datos de menor coste de transmisión o de previsiones de comportamiento.

Finalmente, conviene mencionar los procedimientos de evaluación y seguimiento (López *et al.*, 2002) (Campoy and Sánchez, 2002). Si se pretende utilizar este tipo de entorno en estudios reglados es necesario poder evaluar las actividades que se realizan a través de ellos. Desafortunadamente el hecho que sean actividades no presenciales facilita la picaresca. Una de las soluciones propuestas para evitarla es la generación de un entorno (ej. modelos) diferente para cada usuario, este proceso debe ser automático para dotar al sistema de flexibilidad y eficiencia (Campoy and Sánchez, 2002). No obstante, esta solución, aplicable cuando se trata de prácticas simuladas, pero es de difícil aplicación en entornos con sistemas físicos remotos.

En este artículo se describe la plataforma de experimentación remota que se ha desarrollado en el IOC con el fin de realizar experimentos prácticos remotos en el ámbito de programación de sistemas de control descritos mediante autómatas finitos. El artículo está estructurado de la manera siguiente: en la sección 2 se presentan los objetivos de la plataforma y su ámbito de aplicación; en la sección 3 se describen los elementos que forman el laboratorio remoto; la sección 4 describe la arquitectura de interconexión de los diferentes elementos del sistema; la sección 5 describe las características de la aplicación que se ha desarrollado para realizar la programación remota de los equipos, y finalmente en la sección 6 se describen las principales conclusiones y las ampliaciones previstas para el futuro.

## 2. OBJETIVOS

Las actividades prácticas contempladas en este artículo se enmarcan dentro de la intensificación en Automática de la titulación de Ingeniería Industrial de la ETSEIB y del programa de doctorado en Automatización Avanzada y Robótica del IOC. Ambos estudios incluyen diferentes materias relacionadas con el diseño e implementación de sistemas de control.

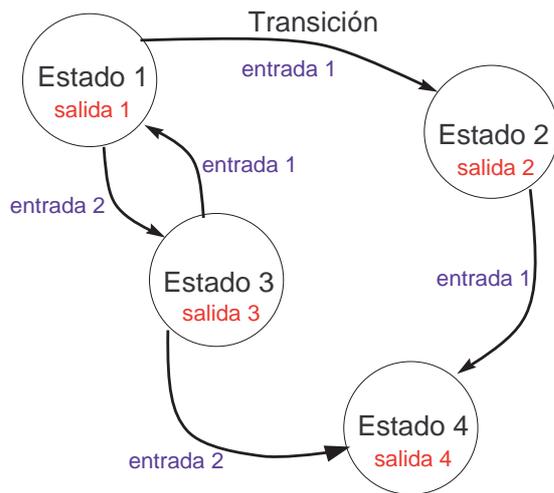


Figura 1. Ejemplo de autómata finito de Moore.

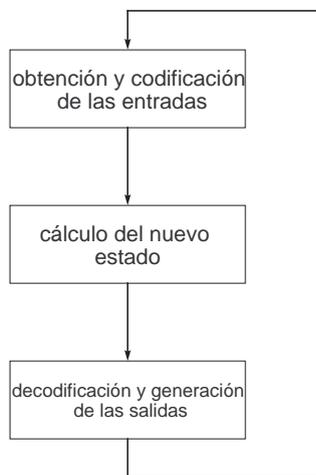


Figura 2. Diagrama de flujo del programa que ejecuta el PLC.

Uno de los componentes más utilizados para la implantación de sistemas de control de forma industrial es el PLC (*Programmable Logic Controller*). Estos dispositivos, cada vez más robustos y flexibles, permiten implementar gran variedad de sistemas de control. Tradicionalmente, han estado dirigidos a sistemas del control de variable discreta, es decir plantas en que tanto los actuadores como los sensores son del tipo todo o nada (es decir, únicamente pueden tomar dos estados, 0 o 1). En la actualidad, la mayoría de PLC's comerciales ofrecen sofisticados entornos de programación. Estos entornos disponen de excelentes interfaces gráficas para codificar, validar y analizar el comportamiento de los programas desarrollados.

Los entornos mencionados suelen aplicar total o parcialmente el estándar IEC 1131-3 que especifica la sintaxis, semántica y visualización de los principales tipos de lenguajes de programación de PLCs. Entre éstos destacan:

- *Lenguaje de Contactos (IEC 1131 Ladder Diagram)*: Este lenguaje utiliza esquemas eléctricos para plasmar el algoritmo deseado. Los elementos son del tipo bobina, condensador, etc.
- *Lenguaje de Instrucciones (IEC 1131 Instruction List)*: Este tipo de programación se realiza directamente en un lenguaje de bajo nivel (en términos de microcontroladores se estaría hablando de lenguaje ensamblador).
- *Lenguaje Estructurado (IEC 1131 Structured Text)*: Este método de programación es el más parecido al método clásico secuencial, con instrucciones del tipo *if*, *while* o *for* entre otras.
- *Grafcet (IEC 1131 Sequential Function Charts)*: Se basa en estados y transiciones entre estados (Figura 1). El estado es la representación, mediante condiciones lógicas, de una situación física real y las transiciones corresponden a las condiciones lógicas que permiten hacer evolucionar el sistema entre los diferentes estados del cual está compuesto.

Tal como se ha descrito, la programación del comportamiento de los PLCs se puede realizar de diversas maneras, para la formación de los estudiantes es importante utilizar un lenguaje formal que, además de describir el comportamiento, permita realizar un análisis formal del mismo. Dentro del ámbito de los sistemas discretos una de estas técnicas regladas son autómata finito (Figura 1). Este formalismo es independiente de la implementación final presenta grandes similitudes con el *Grafcet*. Otro punto a favor de este enfoque es que los comportamientos descritos mediante el mismo pueden ser analizados con métodos formales tales como las Redes de Petri (Silva, 1985). Aunque el objetivo del proyecto es disponer de una herramienta que permita la implementación de una gran variedad de comportamientos y el análisis de los mismos. Se ha optado, inicialmente, por adoptar los autómatas finitos de Moore como herramienta de descripción del comportamiento. Esta opción presenta las ventajas ser fácilmente implementable, requerir una interfaz de usuario sencilla y haber sido estudiada por los estudiantes de ingeniería en las asignaturas de sistemas digitales (ETSEIB, 2004). En contrapartida este formalismo presenta algunas limitaciones, tales como la inclusión de transiciones en función del tiempo o la especificación de comportamientos concurrentes, que serán abordadas en versiones posteriores del sistema.

Aunque la implementación del sistema se podría haber realizado con un PC equipado con los elementos de entrada/salida adecuados, se ha preferido llevarla cabo mediante el uso de PLCs,

dado que son dispositivos robustos y disponen de forma natural de los elementos de entrada/salida necesarios además de ofrecer una buena relación precio/prestaciones. En el laboratorio remoto la robustez es un aspecto especialmente importante pues los equipos han de estar activos durante largos períodos de tiempo.

Aunque desde un punto de vista pedagógico es interesante que los estudiantes desarrollen sus aplicaciones en los entornos de desarrollo del PLC con el que se está trabajando, ya que de esta manera además de adquirir experiencia en el diseño de comportamientos, adquieren también conocimiento de un entorno concreto. Desafortunadamente estos entornos de desarrollo consumen gran cantidad de recursos del sistema, son dependientes de la plataforma de ejecución y son difícilmente distribuibles a través de la red. Otra cuestión que debe tenerse en cuenta es que en su mayoría se trata de entornos propietarios protegidos por licencias y por el secreto industrial, por lo que no se pueden distribuir libremente. Todas estas consideraciones desaconsejan la utilización de dichos entornos para la realización de prácticas remotas.

En el marco de este trabajo se ha optado por el desarrollo de una herramienta ligera, que ocupe poco espacio y pueda ser distribuida libremente. En el momento de plantearla se analizaron dos posibles soluciones:

1. *Desarrollar un entorno propio que permita programar la aplicación de forma similar a como lo hacen los entornos del fabricante.*

El entorno propio debía permitir editar, compilar, validar y enviar al PLC los programas desarrollados. Con esta solución se aprovechan al máximo las prestaciones del PLC y el estudiante puede tener las mismas experiencias que las obtenidas al programar un PLC de la forma tradicional. En contrapartida, implica un importante esfuerzo de desarrollo y está en gran manera ligado a un PLC concreto. Además es necesario obtener del fabricante información de la que actualmente no se dispone y obligaría al estudiante a aprender cuestiones específicas del PLC y del entorno desarrollado.

2. *Desarrollar un programa fijo sobre el PLC que ejecute un autómata finito genérico.*

La implementación del autómata finito se realiza mediante una tabla de transición de estados y una tabla de salidas asociadas a cada estado. Estas tablas se guardan en memoria de datos del PLC y pueden ser cambiadas siempre que sea necesario. El funcionamiento del sistema puede ser alterado sin cambios en el programa que ejecuta el PLC.

La aplicación que debe desarrollarse consta de dos partes, una interfaz de usuario que debe permitir la descripción de los comportamientos y una segunda que ejecute autómatas finitos genéricos. Mientras que la primera de ellas debe ejecutarse en la máquina del usuario la segunda se ejecuta sobre el PLC. En la figura 2 se muestra el diagrama de flujo del programa que ejecuta el PLC<sup>1</sup>.

Este planteamiento es conceptualmente sencillo y fácil de implementar. En contrapartida, no permite aprovechar al máximo las prestaciones del PLC.

Siguiendo el segundo enfoque se ha desarrollado una aplicación que ha permitido validar el planteamiento propuesto.

### 3. COMPONENTES DEL SISTEMA

#### 3.1 PLANTAS

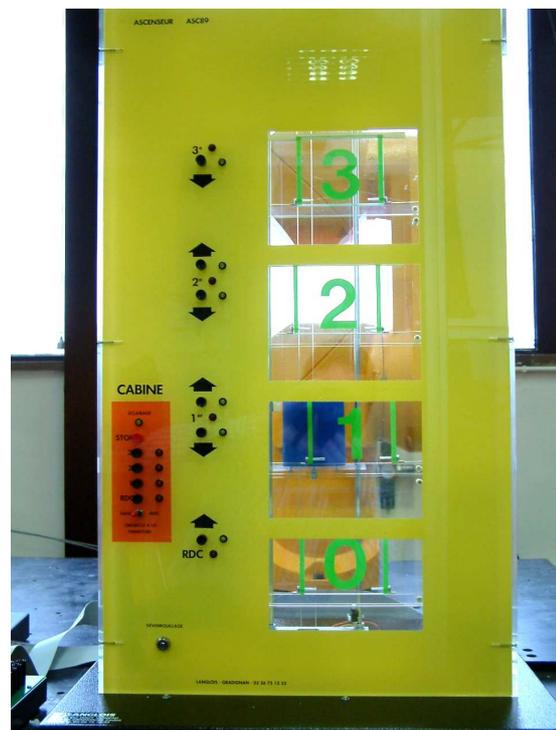


Figura 3. Maqueta de ascensor (ASC89).

Aunque en un futuro se pretende poner a disposición de los estudiantes diferentes tipos de plantas, la puesta en marcha y validación del sistema se ha realizado con una maqueta de ascensor. Pese a lo que podría parecer en un principio, no existen en el mercado muchas maquetas de ascensor. Para el proyecto se ha

<sup>1</sup> Por claridad, en el diagrama de flujos no se ha incluido la gestión de los algoritmos (inicialización, carga de algoritmo, ...)

seleccionado una maqueta de la marca Langlois (Figura 3) que dispone de un total de 24 entradas y 21 salidas. La cabina del ascensor puede acceder a 4 plantas en cada una de las cuales existen pulsadores de llamada para subir y bajar. Pueden también abrirse y cerrarse las puertas de cada planta. Se trata, por tanto de una maqueta con características muy similares a las de un ascensor real y está directamente preparada para interactuar con cualquier tipo de PLC industrial.

Además de la gestión de los diferentes componentes la maqueta de permite especificar comportamientos de mayor complejidad especialmente si se combinan varios ascensores, y por ello resulta de gran interés pedagógico.

### 3.2 PLC

La selección de un PLC es una labor delicada, dado el gran abanico de posibilidades que ofrece el mercado. Dado el carácter didáctico de los equipos que se pretende controlar, el número de entradas y salidas del PLC no es un factor determinante, pues la mayoría de los PLCs del mercado superan los requerimientos necesarios para esta aplicación. Los criterios que se han utilizado para la selección del PLC han sido por tanto, básicamente económicos y de conectividad a Internet. Una opción que ha aportado un equilibrio entre los dos aspectos ha sido el *TSX Micro 37*, ya que, aunque este PLC no dispone directamente de conexión a Internet, es posible conectarlo a un módulo *TSX ETZ 510* que contiene un servidor de web que permite realizar casi cualquier operación sobre el PLC.

En la configuración estándar este PLC dispone de una memoria de 520 palabras de 16 bits cada una. Esta capacidad limitará el tamaño de los autómatas finitos que se podrán implementar. Con esta capacidad de memoria la aplicación desarrollada es capaz de gestionar autómatas de hasta 20 estados y 160 transiciones entre estados. Esta capacidad de representación es en general suficiente para la mayor parte de aplicaciones de carácter didáctico.

### 3.3 SISTEMA DE VÍDEO

Para visualizar los experimentos se utiliza un sistema de vídeo con conexión a Internet formado por un servidor *AXIS 2400+ Video Server* que permite transmitir simultáneamente la señal procedente de 4 cámaras CCD. Este servidor puede ser configurado por software para adaptarlo a las características de la transmisión. Dos de las cámaras pueden ser controladas, es decir el



Figura 4. Sistema de cámaras y servidor multimedia.

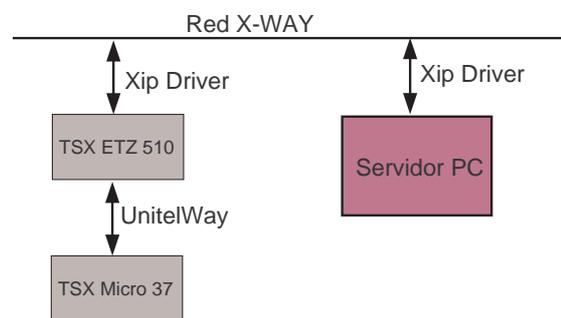


Figura 5. Estructura de la red.

usuario puede desde su aplicación remota modificar la orientación vertical y horizontal de la cámara y el *zoom* para mejorar la vista del experimento. En la actualidad se están utilizando dos cámaras *Canon VC-C4* (Figura 4).

El sistema presenta unas prestaciones ideales en el caso de disponer un ancho de banda importante. Para los casos en el que ancho de banda es insuficiente se ha desarrollado un sistema de monitorización complementario a éste y descrito en la sección 5.

## 4. ARQUITECTURA DEL SISTEMA

### 4.1 Hardware

De forma previa a la implantación del sistema descrito se han analizado diferentes arquitecturas para el sistema (Gasa *et al.*, 2003). El objetivo de incorporar la gestión de experimentos y usuarios a las aplicaciones interactivas ha llevado a seleccionar una arquitectura que incorpore un PC en el sistema, para facilitar el almacenamiento de toda la información necesaria a un bajo coste.

La arquitectura seleccionada está basada en una red *XWAY*. Este tipo de red utilizada por los PLCs de *Schneider Electric*s introduce una dirección universal independiente de la

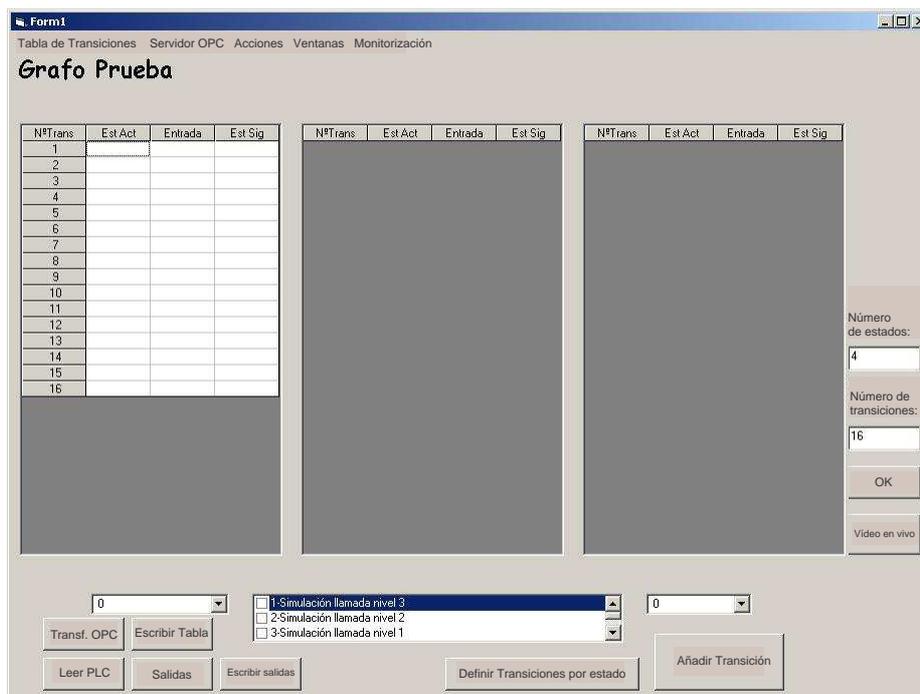


Figura 7. Herramienta de codificación de autómatas finitos mediante tablas de transición

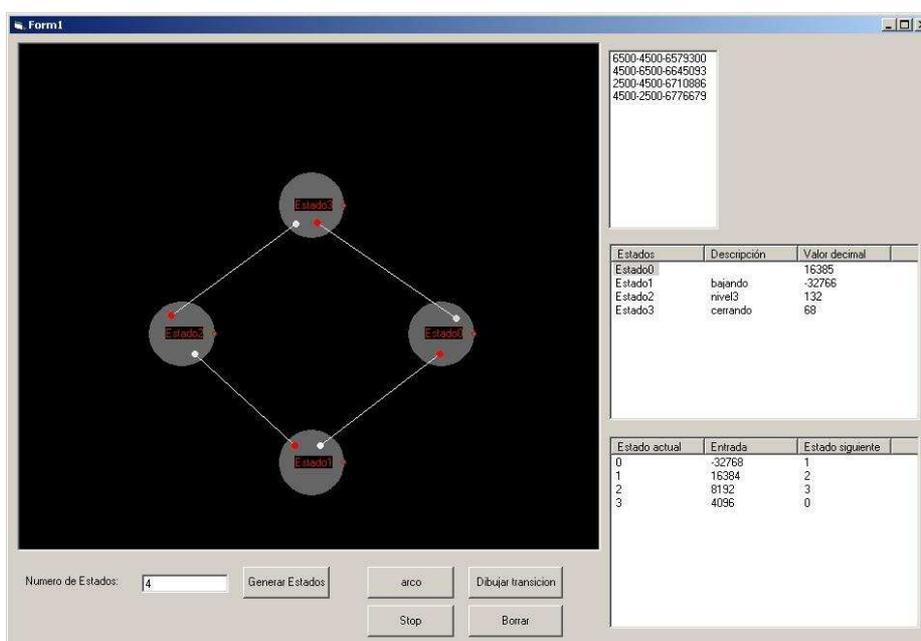


Figura 8. Interfase interactiva de codificación de autómatas finitos

dirección física del dispositivo (ex. TCP/IP, FIPIO, modbus, unitelway), que unifica los diferentes dispositivos y facilita su gestión. En el caso particular de la aplicación descrita el nivel físico corresponde a una red *unitelway* (conexión serie mediante conector RS485). El *host*, PC en nuestro caso, se encarga de gestionar y adaptar las diferentes direcciones. La figura 5 presenta la interconexión de los diferentes componentes de la red.

Con el fin de integrar la gestión de la red en el PC se ha utilizado el *XIP Driver* proporcionado por *Schneider Electric*s. Este *driver* realiza de forma transparente la gestión de direcciones de la red *XWAY*, lo que simplifica la programación del sistema.

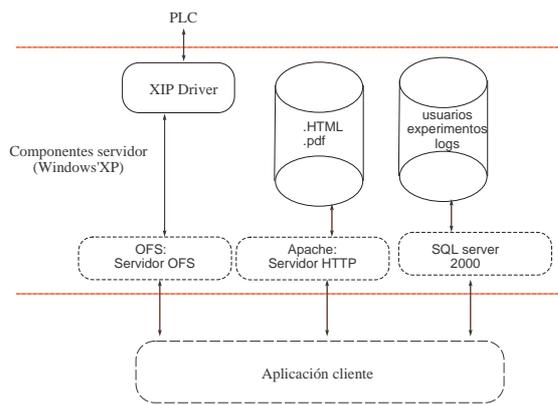


Figura 6. Componentes software del servidor

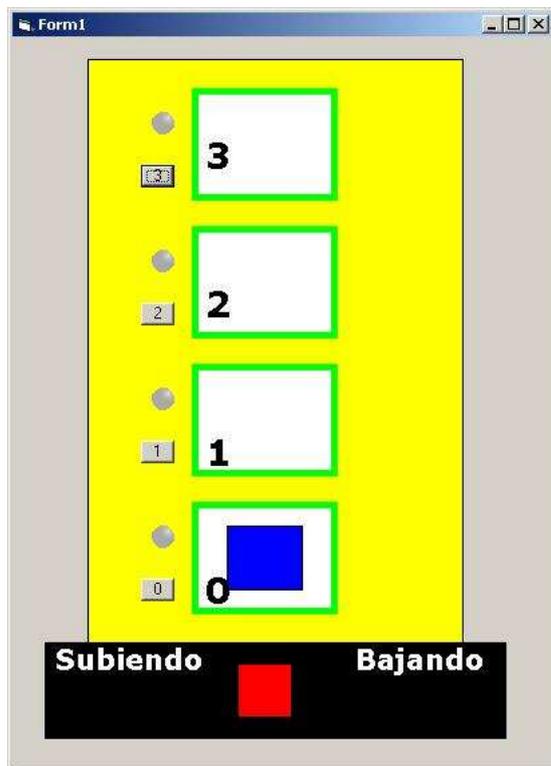


Figura 9. Herramienta SCADA para el seguimiento de la evolución del sistema.

#### 4.2 Software

El *software* del servidor está compuesto por dos componentes una que se ejecuta sobre el PLC y otra que se ejecuta sobre el PC. El *software* que se ejecuta sobre el PLC está implementado en el lenguaje propio del PLC, y su misión es ejecutar el autómata finito genérico (Figura 2). En contrapartida, el código que se ejecuta en el PC es genérico en el sentido que no depende del PLC concreto, y su función es ofrecer conectividad y almacenamiento. La comunicación entre ambos componentes se realiza mediante el protocolo OPC.

Dado que los *drivers* necesarios están disponibles únicamente para plataforma *WindowsXP*, el PC

que actuará como servidor utiliza el sistema operativo *WindowsXP SP1*. Este sistema operativo se complementa con diferentes servidores encargados de ofrecer diferentes funcionalidades. La figura 6 presenta los diferentes elementos implicados.

Se ha incorporado un servidor OPC (*OLE<sup>2</sup> for Process Control*) que permite desarrollar las aplicaciones siguiendo un protocolo industrial basado en estándares abiertos. El protocolo OPC permite acceder a diferentes dispositivos industriales de una manera uniforme, lo que facilita el desarrollo de aplicaciones de supervisión y control. La mayoría de lenguajes de programación de alto nivel incorporan librerías para acceder de forma sencilla a este protocolo. Además de la facilidad de programación introducida, el servidor OPC es el encargado de la gestión de las diferentes conexiones que se pueden producir en el sistema.

Dado que el servidor OPC actúa de forma distribuida y en un computador con sistema operativo *WindowsXP* deberá registrarse en los componentes distribuidos DCOM (*Distributed Components*) y ser configurado de acuerdo con las características de la aplicación. El servidor OPC que se utiliza es el OFS (*OPC Factory Server*) de *Schneider Electric*s.

Con el fin de hacer un seguimiento de los diferentes experimentos y conexiones, y tener un control de los usuarios del sistema se ha introducido la base de datos *SQL server 2000* encargada de gestionar dicha información.

Finalmente se ha incorporado un servidor HTTP (Apache) con el fin de distribuir la aplicación cliente y los diferentes documentos de información y los enunciados de las prácticas.

#### 5. APLICACIÓN CLIENTE

La aplicación cliente se ha desarrollado íntegramente en *Visual Basic*, lo que ha permitido disponer de forma rápida de una aplicación para realizar los experimentos deseados. La comunicación con el servidor OPC se ha implementado mediante la librería de libre distribución OPC Automation 2.0.

La aplicación permite especificar los grafos de comportamiento mediante tablas de transición (Figura 7) o dibujando el grafo directamente a través de una interfase interactiva (Figura 8). Aunque la representación gráfica resulta más atractiva para los estudiantes, se ha considerado de interés pedagógico generar las tablas de transición y salidas pues de esta manera los estudiantes pueden captar la relación entre ambas representaciones (la aplicación genera de forma

<sup>2</sup> Object Linking and Embedding

automática la tabla de transiciones a partir del autómata finito). Con el fin de facilitar la programación se ha asignado nombres lógicos a las diferentes variables físicas y a los valores que estas pueden tomar.

Una vez se ha especificado el comportamiento mediante alguno de los métodos indicados, la propia aplicación se pone en contacto con el servidor e inicia el proceso de carga y ejecución remota del programa diseñado. La aplicación incorpora la posibilidad de seguir la evolución del sistema mediante las imágenes generadas en vivo por el servidor *AXIS 2400+ Video Server*<sup>3</sup>. Desafortunadamente esta forma de seguir la evolución del sistema únicamente es viable si se dispone de una conexión de altas prestaciones. Por dicho motivo se ha desarrollado un módulo SCADA (*Supervisory Control And Data Acquisition*) que permite visualizar la evolución de ascensor a partir de la información recibida del servidor OPC y de una representación gráfica simplificada (Figura 9). Este módulo permite la interacción con la planta física (ej. hacer que el ascensor vaya a una cierta planta ...) y podría considerarse como un caso sencillo de realidad aumentada.

Además de las funcionalidades descritas, la aplicación cliente interacciona con la base de datos del servidor con el fin de validar al usuario, almacenar los diferentes experimentos y obtener información sobre la planta concreta con la que se pretende trabajar.

## 6. CONCLUSIONES

En este artículo se ha descrito la arquitectura hardware/software que se ha utilizado en el IOC para la realización de prácticas de programación de PLCs de forma remota. La arquitectura y herramientas utilizadas han permitido poner en marcha el sistema con esfuerzo moderado. La plataforma desarrollada además de servir para especificar el comportamiento del sistema mediante autómatas finitos de *Moore* permite monitorizar su ejecución ya sea mediante una transmisión de vídeo o un sistema SCADA.

El trabajo realizado constituye un primer paso que ha permitido validar la arquitectura y los diferentes conceptos presentados. En la actualidad se está validando una nueva versión en la que la aplicación cliente está plenamente implementada en JAVA (Garrido, 2003). Ello permite que la aplicación cliente sea independiente de plataforma.

Líneas futuras de trabajo son la incorporación de herramientas de soporte al análisis y diseño (ej. Redes de Petri). En concreto se pretende incorporar un simulador en la aplicación de usuario para que éste pueda validar el comportamiento del sistema de forma previa a la ejecución en la planta real. De forma similar se pretende validar la arquitectura con otros tipos de plantas didácticas tales como un robot móvil.

## Agradecimientos

Los autores quieren expresar su agradecimiento a *Schneider Electric*s por el soporte prestado durante el desarrollo de este trabajo, en especial a Joan Jimenez por su soporte y colaboración.

Este trabajo ha sido parcialmente financiado por los proyectos "Laboratorios Remotos de Automática (AFC2001-0388-LD)" y "Laboris Remots de Automàtica (UPC)".

## REFERENCIAS

- Azuma, R. T. (1997). A survey of augmented reality. In *Presence: Teleoperators and Virtual Environments* 6(4), 355–385. Available online.
- Campoy, P. and C. Sánchez (2002). Ejercicios personalizados de control de sistemas reales simulados. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- ETSEIB (2004). *Guia Docente. Escuela Técnica Superior de Ingeniería Industrial de Barcelona (ETSEIB)*.
- Garrido, Ignacio (2003). Maqueta de ascensor para la realización de prácticas por internet. Master's thesis. ETSEIB.
- Gasa, D., I. Garrido, R. Costa and L. Basañez (2003). Laboratorio remoto de automática: plantas de variable discreta. In: *Actas de las XXIV Jornadas de Automática*. CEA-IFAC. Universidad de León.
- Giralt, X., D. Jofre, R. Costa and L. Basañez (2002). Proyecto de laboratorio remoto de automática: Objetivos y arquitectura propuesta. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- Guzmán, J.L., J.F. Sarabia, F. Rodríguez, J.C. Moreno and M. Berenguel (2002). Entorno para programación remota de robots manipuladores en acl con realimentación visual. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- López, J. A., E. Gómez, J.L. Risco, B. Andres-Toro and J.M. de la Cruz (2002). Sistema

<sup>3</sup> La incorporación de estas imágenes en la aplicación es especialmente sencilla debido a que el propio fabricante proporciona los controles *ActiveX* necesarios para ello.

- automático de evaluación de conocimientos. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- Maza, J.I., A. Castro and A. Ollero (2002). Hetero v2.0: Herramienta para la tele-enseñanza de la robótica. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- McKee, G. T. (2002). Online robot educational environments. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- Puerto, R., L. M. Jiménez and O. Reinoso and C. Fernández (2002). Recolab: laboratorio de prácticas de control de procesos vía internet. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- Sánchez, J., F. Morilla and S. Dormido (2001). Teleoperation of an inverted pendulum through the world wide web. In: *Workshop On Internet Based Control Education. IBCE'01*. Madrid, Spain. pp. 12–14.
- Sebastián, J. M., F. M. Sánchez, D. García and R. Aracil (2002). Sivanet: un nuevo escenario físico remoto para el auto-aprendizaje de control a través de internet. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- Serrano, C., R. Costa and L. Basañez (1999). VRML-JAVA aplicaciones a la programación fuera de línea de robots industriales. In: *6to Congreso de la Asociación Española de Robótica y Automatización Tecnologías de la producción*. AER. Barcelona. pp. 20–22.
- Silva, M. (1985). *Redes de Petri en la Automática y la Informática*. Editorial AC.
- Torres, F., F. A. Candelas, J. Pomares, S. T. Puente, F. G. Ortiz and P. Gil. (2002). Laboratorio virtual remoto para la enseñanza de robótica. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.
- Vallés, M., A. Valera and J.L. Díez (2002). Desarrollo y control de procesos virtuales mediante matlab web server. In: *Actas de las III Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante.