

SIMULACIÓN REMOTA DE PROCESOS Y SISTEMAS DE ENTRADA/SALIDA

Carlos Domínguez, Houcine Hassan, Ángel Perles
Departamento de Informática de Sistemas y Computadores
Universidad Politécnica de Valencia
{carlosd, husein, aperles}@disca.upv.es

Resumen

Se presenta simPROCes, una arquitectura de simulación para la validación de sistemas informáticos de control, desarrollada para la docencia de la informática industrial, y que es aplicable también a la validación de sistemas reales.

simPROCes permite simular el proceso bajo control junto con el sistema de entrada/salida que le sirve de interfaz, de forma que cargando distintos modelos de proceso y de entrada/salida, se tienen variados sistemas de prueba.

El simulador del proceso y la aplicación de control que se pretende validar, se comunican mediante TCP/IP, lo que permite ubicar cada módulo donde se desee en la red. Dicha conexión, no obstante, pasa desapercibida para el controlador, que usa la misma interfaz del sistema de entrada/salida real que se pretende instalar en la versión final.

Actualmente está desarrollada, como un ejemplo, la simulación de la tarjeta de entrada/salida ADLink PCI9112 [2] y la de un depósito de agua, elementos reales usados en el laboratorio.

Palabras Clave: Simulación Remota, Modelo de Proceso, Modelo de Entrada/Salida, Laboratorio Virtual.

1 INTRODUCCIÓN

La validación es clave en el desarrollo de sistemas informáticos industriales, y los simuladores de

procesos físicos son una herramienta económica para realizar las pruebas.

Una parte importante de la actividad de los alumnos de Informática Industrial en la Escuela Universitaria de Ingeniería Técnica Industrial de Valencia consiste en la realización de un miniproyecto informático de control [3].

Para facilitar las pruebas de los sistemas que los alumnos realizan, el grupo de profesores está desarrollando herramientas de simulación de bajo coste.

En una primera fase se desarrolló, y se ha utilizado ya durante un par de cursos, el entorno SimSeny [4], un simulador de la tarjeta de entrada/salida ADLink PCI9112 instalada en el laboratorio.

SimSeny ha permitido a los alumnos simular la entrada/salida de señales, trabajando sobre cualquier ordenador. De esta forma no han estado obligados por el apretado horario del laboratorio, único espacio equipado con las tarjetas reales.

Con esta herramienta es posible ejercitarse en la programación de la entrada salida, pero al no contemplar el modelo del proceso bajo control, las señales deben generarse manualmente, resultando poco práctico, y en muchos casos imposible.

Para incorporar el modelo del proceso, en una segunda fase, se ha desarrollado simPROCes, que comenzará a utilizarse durante el próximo curso.

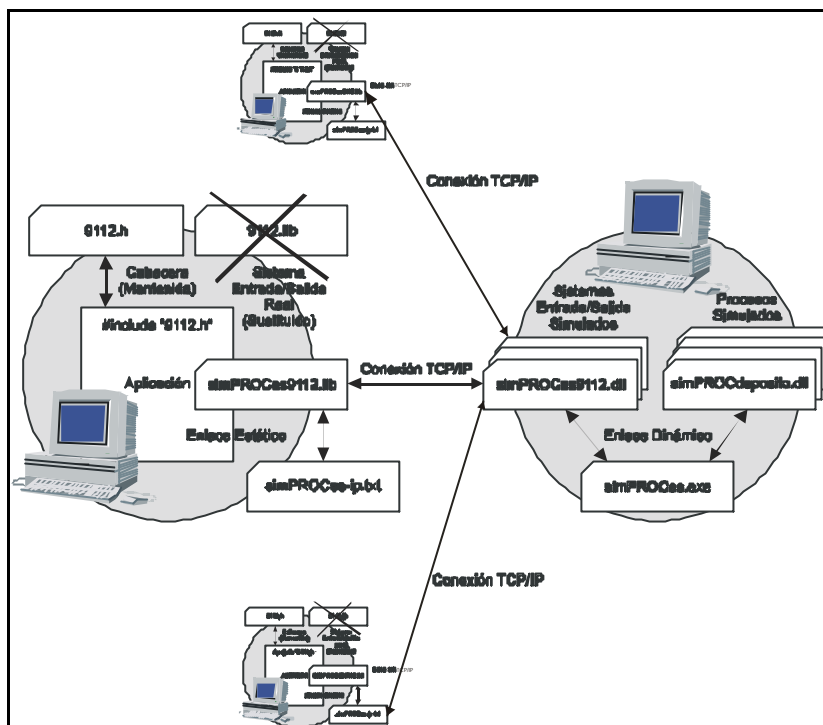


Figura 1. Arquitectura de simPROCes

Los requerimientos para simPROCes han sido:

- Simular tanto el proceso como el sistema de entrada/salida.
- Definir un entorno que permita cargar distintos modelos.
- Simular transparentemente desde el punto de vista del programa de control.
- Ubicar el simulador en cualquier sitio de la red.

El resto del artículo describe brevemente simPROCes y su uso. En el apartado 2 se muestra la arquitectura y las interfaces que deben respetar los programadores de módulos de simulación para simPROCes. En el apartado 3 se muestra un ejemplo de uso indicando los pasos que debe seguir el programador de la aplicación de control. Finalmente, en el apartado 4 se resumen las conclusiones y se expone el proyecto futuro.

2 ARQUITECTURA simPROCes

El laboratorio virtual que permite la validación de controladores se puede apreciar en la figura 1. Su arquitectura se compone de un servidor de modelos de procesos (derecha de la figura 1) y de un conjunto de clientes (izquierda) que se desarrollan para validar sus controladores sobre el servidor. Aunque en la configuración típica, el servidor y el cliente se ejecutan sobre ordenadores distintos, también es posible ejecutar ambos componentes sobre una

misma máquina, aprovechando la dirección IP de autobucle 127.0.0.1.

2.1 LOS MÓDULOS

El módulo que define la arquitectura es el programa simPROCes.exe (en la parte inferior derecha del gráfico), que permite al usuario la carga en tiempo de ejecución de una librería con el simulador del proceso y de otra con el simulador del sistema de entrada/salida, ambas a elección de entre los modelos que puedan estar desarrollados.

simPROCes.exe es el nexo entre el proceso y el sistema de entrada/salida, definiendo la asignación de puertos a señales físicas de accionamiento y de sensorización.

El simulador del sistema de entrada/salida, se realiza con dos módulos enlazados por una conexión TCP/IP, lo cual permite tanto la ubicación de los componentes en una sola máquina, como en diversas máquinas conectadas en la red. Uno de dichos módulos es la librería cargada por simPROCes, el otro es la librería que debe enlazarse con el programa de aplicación.

Desde el punto de vista del programador de la aplicación de control (a la izquierda del gráfico), el uso del simulador afecta sólo en el proceso de montaje, cuando se sustituye un módulo de librería

facilitado por el fabricante de la tarjeta de adquisición de datos por el módulo simulado. El proceso de traducción no se altera, ya que se sigue empleando el mismo archivo de cabecera, ofreciendo el simulador de entrada/salida una interfaz equivalente al sistema real.

En simPROCes se intenta validar la aplicación para distribuirla tal cual como se tradujo, por tanto, la información relativa al funcionamiento del simulador, como por ejemplo la ubicación del simulador del proceso, se consigue abriendo el archivo de texto simPROCes-ip.txt de forma totalmente transparente.

2.2 LAS INTERFACES

La arquitectura simPROCes define las interfaces que deben respetar las librerías de enlace dinámico que pretendan conectarse a él, para simular procesos o sistemas de entrada/salida.

Así, la figura 2 muestra la interfaz que los módulos de simulación de proceso deben exponer a simPROCes.exe.

```
int interrogar_p(char *facilidad);
int parametrizar(double *pars);
int iniciar(double *estado);
int parar(void);
int continuar(void);
int accionar(double *accs);
int perturbar(double *perts);
int observar(double *obs);
int informar(int *opciones);
```

Figura 2: Interfaz del simulador de proceso

Seguidamente se comentan brevemente dichas funciones.

La función de interfaz `interrogar_proceso()` permite a simPROCes determinar las características básicas del simulador del proceso, relativas a las señales de accionamiento y de sensorización, así como a conjuntos seleccionados de variables de estado y de parámetros de configuración.

`parametrizar()` permite configurar el sistema modelado. Por ejemplo, la figura 3, que muestra la vista de la realización `simPROCdeposito.dll` del depósito de agua del laboratorio, marca en negro ciertos parámetros ajustables, como la superficie de la base o las constantes características de las válvulas. La parametrización permite ensayar un

conjunto de casos a partir del mismo esquema de proceso.

`iniciar()`, `parar()` y `continuar()`, controlan el proceso de simulación. La primera de las funciones acepta un estado inicial de partida.

`accionar()` y `perturbar()` provocan las excitaciones en el proceso. Las acciones vienen, a través del simulador de entrada/salida, desde la aplicación de control; mientras que las perturbaciones son generadas desde el propio simPROCes.

`observar()` traslada las variables observables de los sensores, de nuevo a través del simulador de entrada/salida, hacia el controlador.

Finalmente, `informar()` permite desplegar una representación gráfica del estado del proceso que el propio simulador ofrece. Aunque los alumnos construyen una interfaz gráfica con el operador en su aplicación de control, les ayuda disponer de estas vistas ya en la fase de especificación.

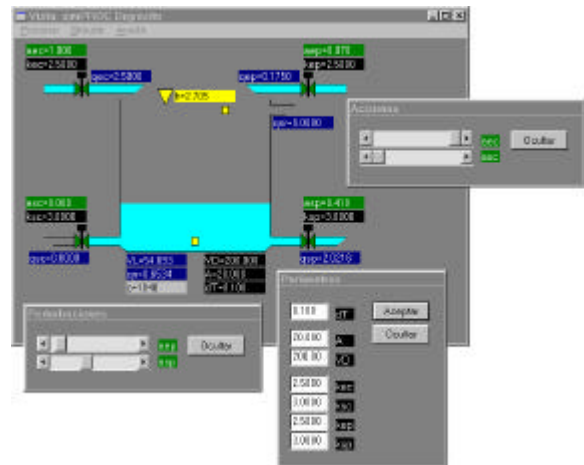


Figura 3: Vista del simulador del depósito de líquido

Análogamente, en la figura 4 puede verse la interfaz para los módulos de entrada/salida.

```
int interrogar_es(char *facilidad);
int conectar(int nent,int *ent,
            int nsal,int *sal);
int entrar(double *parametros);
int salir(double *estado);
int manipular(int *opciones);
```

Figura 4: Interfaz del simulador de entrada/salida

La función `interrogar_es()` permite a `simPROces` determinar las características básicas del simulador de entrada/salida, relativas a los tipos de señales que puede manejar.

`conectar()` describe los puertos de entrada a utilizar para observar los sensores y los puertos de salida a utilizar para manipular los accionadores.

`entrar()` y `salir()` realizan las observaciones y los accionamientos.

Finalmente, `manipular()` permite desplegar un conjunto de controles para ver y/o modificar las señales implicadas, como muestra la figura 5.

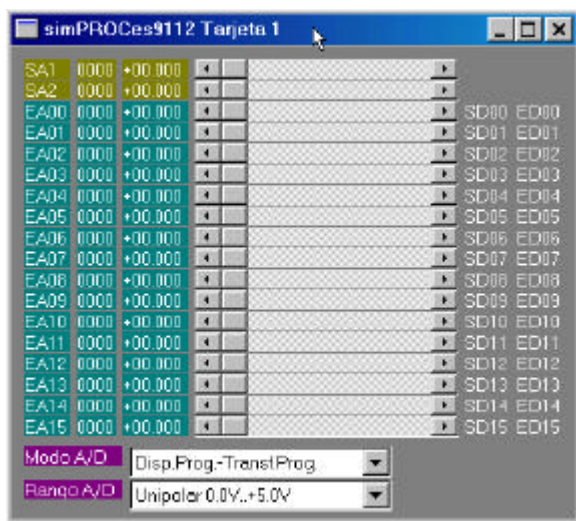


Figura 5: Vista de un simulador de entrada/salida

2.3 LAS FACILIDADES

Tanto las facilidades que ofrece el proceso como las del sistema de entrada/salida se exponen sobre un conjunto de tablas predefinido (actualmente sólo en formato Paradox) según muestra la tabla 1.

Archivo	Contenido
E.DB	Entradas
S.DB	Salidas
OM.DB	Opciones de Manipulación

Archivo	Contenido
P.DB	Parámetros
VE.DB	Variables de Estado
VO.DB	Variables Observables
VC.DB	Variables Controlables
VP.DB	Variables de Perturbación
OI.DB	Opciones de Información

Tabla 1: Tablas de facilidades

La figura 6 muestra un formulario con las facilidades del proceso del depósito de agua, que puede ser inspeccionado desde el entorno.

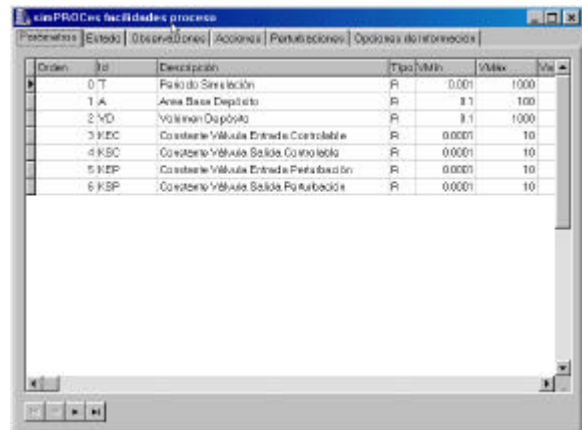


Figura 6: Ejemplo de formulario de facilidades

3 EJEMPLO DE USO

Se describe en este apartado las modificaciones que debe realizar el programador de la aplicación de control para la construcción de una versión de prueba; modificaciones respecto la versión final. Asimismo, se indican los pasos necesarios para iniciar la simulación.

3.1 CONSTRUCCIÓN DE LA APLICACIÓN DE CONTROL PARA PRUEBAS

Desde el punto de vista del programador de la aplicación de control, obtener una versión de prueba consiste sólo en sustituir durante el montaje la librería de entrada/salida que proporciona el fabricante de la tarjeta real por la librería simulada.

Al proporcionar el simulador la misma interfaz que el sistema real, la codificación y la traducción del programa de control no resulta alterada.

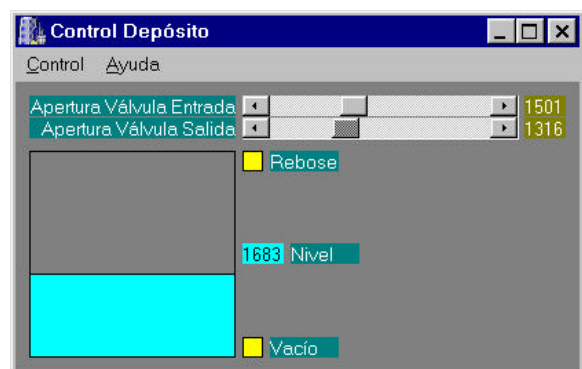


Figura 7: Vista de un controlador de demostración

La figura 7 muestra un programa de demostración para el control del depósito que se ofrece con el entorno. Está construido sobre Borland C++ Builder 5 [1], la herramienta de programación empleada por los alumnos.

Dicha aplicación realiza el control y monitorización del nivel del líquido en el depósito. El programador crea una aplicación (ver figura 1) donde implementa el controlador que quiere probar y mediante las funciones del interfaz con la entrada/salida y la conexión a TCP/IP puede interactuar, manipulando las válvulas de entrada y salida, con el proceso del depósito dispuesto en la máquina servidora.

Al simularse la tarjeta PCI-9112, la adquisición de datos se realiza invocando funciones que presentan la misma cabecera que las ofrecidas por la librería del fabricante.

Así, por ejemplo, para iniciar la tarjeta y establecer modos y rangos de operación se invocan las funciones:

```
W_9112_Initial()  
W_9112_AD_Set_Mode()  
W_9112_AD_Set_Range()
```

para realizar las entradas digitales de los sensores de REBOSE y VACÍO:

```
W_9112_DI()
```

para realizar salidas analógicas de manipulación de las válvulas de ENTRADA y SALIDA:

```
W_9112_DA()
```

y para realizar la entrada analógica desde el sensor de NIVEL:

```
W_9112_AD_Set_Channel()  
W_9112_AD_Soft_Trig()  
W_9112_AD_Aquire()
```

3.2 PROCESO DE SIMULACIÓN

Una vez construida la aplicación de control en su versión de prueba, para validarla, debe iniciarse previamente el simulador.

Si el programador dispone del entorno simPROCes y de las librerías con los modelos de proceso y sistema de entrada/salida, puede ejecutar tanto el controlador como el simulador sobre el mismo ordenador. Para ello, antes de iniciar la simulación, debe escribir la dirección de autobucle 127.0.0.1 en el archivo de

texto simPROCes-ip.txt que automáticamente será consultado al llamar la función de inicialización de la tarjeta desde el controlador.

Si por el contrario, no dispone de los modelos, debe solicitar al propietario de los mismos la dirección IP de un ordenador donde haya una instancia del simulador iniciada. Es el caso en el que los profesores facilitan a los alumnos la dirección IP de un ordenador del laboratorio que actúa como servidor de modelos.

Los pasos se resumen pues en:

- Iniciar una instancia del simulador
- Editar la dirección IP del simulador
- Iniciar el controlador de prueba

Por su parte, el inicio de una instancia de simulación debería comprender las siguientes actividades:

- Iniciar el entorno simPROCes
- Cargar un modelo de proceso
- Cargar un modelo de sistema de entrada/salida
- Cargar un esquema de conexiones
- Parametrizar e iniciar la simulación

4 CONCLUSIONES

En este artículo se ha presentado simPROCes, una herramienta que permite validar controladores de procesos de forma remota. Las ventajas de disponer de dicha herramienta son por una parte, que el hecho de que se puedan realizar pruebas sobre modelos de procesos antes de manipular los prototipos reales permite reducir los costes derivados de la mala utilización que se pueda hacer de los mismos. Por otra parte, trabajar con enlaces de red de forma remota con los procesos permite mayor flexibilidad en los horarios y menor necesidad de espacios en los laboratorios.

La versión actual de simPROCes cuenta con un simulador para la tarjeta de adquisición de datos ADLink PCI9112 y para el proceso del depósito de agua. Durante este próximo curso se desarrollarán otros modelos de procesos, sobre los temas que tradicionalmente se han empleado en las asignaturas de informática industrial, como las expendedoras, los ascensores, etc. [5].

La carga de la red dificulta la simulación de procesos si son de dinámica rápida. Como siguiente paso, se pretende incorporar una interfaz de gestión del tiempo, para permitir simulaciones con dilatación/contracción temporal.

La herramienta que se ha presentado en este artículo está disponible para su uso en el servidor de las asignaturas [5].

Referencias

- [1] Borland C++ Builder 5.0
© 2000 Inprise Corporation
<http://www.borland.com>

- [2] PCI-9112
PCI-Bus Advanced Data Acquisition Card
(C/C++ & DLL Library)
© 1999 ADLink Technology Incorporated
<http://www.web-tronics.com>

- [3] Perles; Domínguez; Martínez; Hassan;
"Enseñanza de la Informática Industrial mediante Proyectos Reales Simplificados".
La Almunia de Doña Godina: Jornadas de la Enseñanza Universitaria de la Informática.
1999. ISBN 84-605-9617-6 Páginas 55 a 60

- [4] Perles; Martínez; Albaladejo; Domínguez;
"SimSeny: Un Simulador Didáctico para Tarjetas de Adquisición de Datos". IV Congreso de Tecnologías aplicadas a la enseñanza de la electrónica. Barcelona 2000.
ISBN 84-600-9596-7 Páginas 493 a 496

- [5] Servidor de las asignaturas de informática industrial. <http://infoind.disca.upv.es>