SimProc, una herramienta de simulación de procesos
Manual de usuario

Óscar Gómez García

7 de abril de 2007

Índice

Índice de figuras

1. ¿Qué es SimProc?

SimProc es una herramienta de simulación de procesos. Está construido para el apoyo a la enseñanza y permite a los alumnos construir sistemas que automaticen procesos discretos, es decir, procesos en los que el estado depende de variables con valores discretos tales como las salidas de pulsadores (salidas todo/nada) o biestables. SimProc permite construir sistemas y manipularlos de forma interactiva para ver el resultado del diseño de un sistema y comprobar su correcto funcionamiento antes de ser implantado.

SimProc forma parte de una serie de desarrollos realizados en la Escuela Superior de Informática de Ciudad Real, encaminados a la consecución de un conjunto de herramientas software que faciliten o abaraten tareas de automatización. En la actualidad existen multitud de simuladores de procesos de gran valor didáctico pero cuyo coste los hace inasequibles para los presupuestos de entornos académicos: SimProc es un software libre basado en tecnologías abiertas que permite al profesorado y al alumnado realizar dichas tareas y compartir los sistemas diseñados de una forma muy sencilla.

2. Instalación.

Se proporciona un instalador para Windows que automatiza el proceso de instalación. El mecanismo de instalación es el clásico para estos entornos y la única necesidad de interacción por parte del usuario se reduce a indicar el directorio de instalación.

No se requiere ninguna instalación en entornos Linux si no se desea. Se proporciona un ejecutable autocontenido con todas las bibliotecas que usa el programa enlazadas de forma estática. Para su uso en estos entornos basta con ejecutar el fichero desde un entorno X-Window.

3. Compilación.

En caso de que se desee modificar, ampliar o simplemente recompilar la aplicación existen una serie de pasos a dar a la hora de compilar la aplicación. Tales pasos son diferentes en función de los entornos donde se desee compilar.

3.1. Compilación en entornos Windows.

A continuación se detallan los pasos a dar en caso de que se desee recompilar SimProc para un entorno Windows:

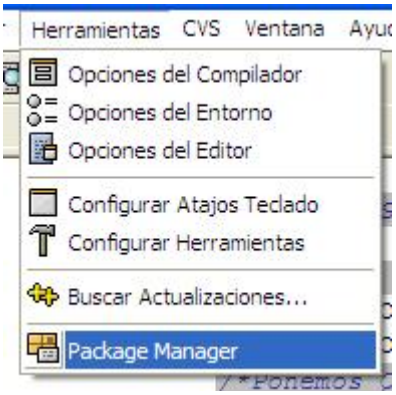

Figura 1: Añadiendo nuevas bibliotecas para el desarrollo desde Dev-Cpp

3.1.1. Preparación de las herramientas

1. SimProc ha sido desarrollado principalmente con Dev-Cpp, un IDE para el compilador GCC que permite desarrollar aplicaciones Windows. Dev-Cpp es software libre y se puede descargar desde el sitio oficial de Dev-Cpp por lo que el primer paso para la recompilación pasa por descargar e instalar este entorno.

2. Se necesita descargar también Xerces, la biblioteca del proyecto Apache para la manipulación de XML. Se puede obtener más información y descargas desde la página del proyecto XML de Apache. Una vez descargada se debe descomprimir en un directorio cualquiera al cual se hará referencia más tarde.

3. Descargar el DevPak de WxWidgets para Dev-Cpp. Un Dev-Pak es una biblioteca empaquetada para uso desde el entorno de Dev-Cpp. Una vez descargado desde http://devpaks.org se pueden incorporar a Dev-Cpp desde el menú Herramientas->Package Manager tal y como se muestra en la figura 1.

3.1.2. Compilación del proyecto

Se proporcionan ficheros de proyecto con todo preparado para que la compilación se reduzca a una simple pulsación de ratón. En caso de problemas se puede construir un nuevo proyecto que recompile la aplicación y para ello se puede hacer lo siguiente.

- Se puede recompilar la biblioteca que maneja los elementos y el XML que los representa entrando en el directorio `libreriaXML` y construir un

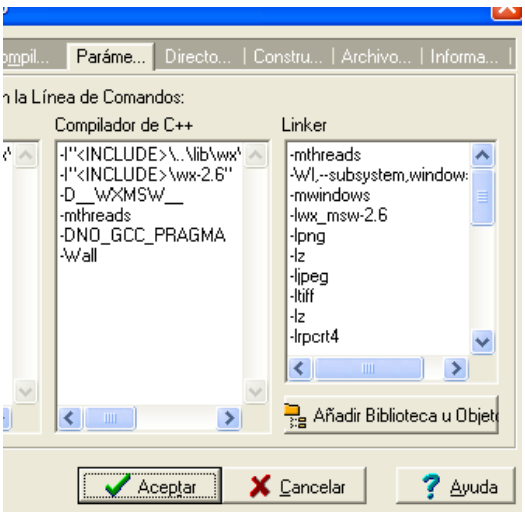

Figura 2: Enlazado de bibliotecas

nuevo proyecto vacío al cual se añadirán todos los ficheros .cpp y .h que haya en este directorio. Una vez añadidos se debe indicar al compilador que enlace con la biblioteca de Xerces en el menú Proyecto->Opciones de proyecto. Al abrir dicho menú se abre una ventana donde se podrá ver la solapa "Parámetros". Utilizando el botón "Añadir biblioteca u objeto" se podrá indicar la ruta hacia la biblioteca `xerces.lib`. En la figura 2 se muestra la apariencia de dicha ventana.

- El entorno gráfico de la aplicación se puede recompilar desde cero creando un proyecto vacío de WxWidgets al cual se añadirán todos los ficheros .cpp y .h que hay dentro del directorio `interfaz` al cual se añadirá luego la referencia a la biblioteca estática que se ha creado en el paso anterior y de la misma forma que muestra la figura 2.

3.2. Compilación en entornos Linux.

La compilación de la aplicación para un entorno Linux es bastante más sencilla.

1. Descargar el código fuente de Xerces y compilarlo. Las opciones por defecto son suficientes para obtener una versión funcional de la aplicación. Se puede instalar con los tres pasos típicos de las herramientas GNU:

a) `./configure`

b) `make`

c) `make install`

2. Compilar WxWidgets: Aquí existen multitud de opciones ya que se puede compilar WxWidgets enlazando con el toolkit de X, con el GTK o con muchos otros como el de Macintosh. En las capturas de pantalla que se muestran aquí, se puede ver SimProc enlazado con el toolkit GTK. De igual forma que en el paso anterior, compilaremos e instalaremos WxWidgets con

   a) `./configure`

   b) `make`

   c) `make install`

3. Por último solo queda compilar la aplicación. Dado que no se necesita ninguna característica extra que no ofrezcan Xerces o WxWidgets, este paso se reduce a ejecutar el archivo `compilar.sh` que se encuentra en el directorio raíz del código fuente. Este script entrará en el directorio denominado `./libreriaXML` y construirá la biblioteca que manipula los elementos y su XML y después compilará el interfaz gráfico. Se producirá un archivo llamado SimProc que debe ser ejecutado desde un entorno gráfico como KDE o GNOME.

4. El entorno de SimProc.

Una vez arrancado SimProc se obtiene una ventana como la mostrada en la figura 3 (si se abre en un entorno Linux la ventana es muy similar, tal y como se puede ver en la figura 4)

En la aplicación hay cuatro grandes zonas cuyo uso y funcionamiento se detallan a continuación. En primer lugar hay una barra de herramientas a la izquierda que contiene los elementos que se pueden usar para construir sistemas y que se puede ver en la figura 5

En la columna de la izquierda y de arriba a abajo podemos encontrar los siguientes elementos:

Pulsador: produce un voltaje al ser pulsado.

Electroimán: produce un campo magnético al ser alimentado.

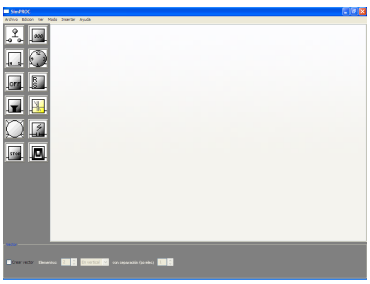

Figura 3: La ventana de SimProc

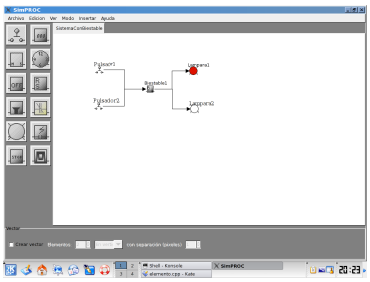

Figura 4: El entorno de SimProc en KDE

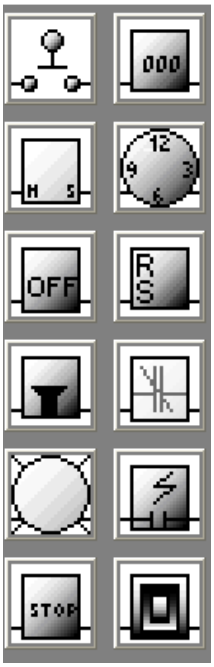

Figura 5: Barra de herramientas de SimProc

Sensor reed: produce un voltaje de salida cuando detecta un campo magnético.

Zumbador: produce un zumbido al ser alimentado.

Lámpara: produce luminosidad al ser alimentado.

Motor: produce un giro en sentido horario o antihorario en función de la alimentación.

En la columna de la derecha y de arriba a abajo se pueden encontrar los siguientes elementos:

Contador: cuenta el número de flancos y produce un voltaje cuando se alcanza la cuenta máxima. Un flanco es un cambio en la entrada de una alimentación positiva a una nula o a la inversa. Por ejemplo, cuando la alimentación pasa de ser 0 a ser 5 (los valores por defecto) se produce un flanco.

Temporizador: espera un cambio en la entrada y cuando se produce el cambio espera un determinado número de segundos para lanzar un cambio en su salida. Permite activar elementos esperando un cierto tiempo.

Biestable: tiene dos salidas denominadas"Q" y "Q negada", siendo una la inversa booleana de la otra. El biestable tiene dos entradas denominadas "R" y "S". Si se produce un flanco de subida en la entrada R, el biestable produce en su salida "Q" el voltaje correspondiente a un 0 lógico y un 1 lógico en su salida "Q negada". Si se produce el flanco de subida en "S" se obtendrá un 1 lógico en "Q" y un 0 lógico en "S". El biestable "recuerda" el valor de las salidas "Q" y "Q negada" aunque se reinicialicen las entradas.

Relé: produce un voltaje de salida en función del voltaje a la entrada. Los relés permiten construir toda una lógica de control así como separar la lógica de control de los elementos de amplificación de voltaje. Se verá un ejemplo más detallado en las secciones siguientes.

Fotosensor: produce un voltaje de salida si detecta una luminosidad. A fecha de hoy, los fotosensores incorporados en SimProc solo son sensibles a una cierta tonalidad de rojo, en concreto la que producen por defecto las lámparas de SimProc.

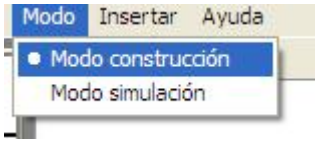

Figura 6: El menú modo

Sistema: permite insertar un subsistema que nos permitirá seguir construyendo un sistema a base de piezas más pequeñas.

El uso de subsistemas es la carácterística más potente de SimProc ya que permite dos cosas:

1. Por un lado, permite hacer diseños de forma muy sencilla al permitir la construcción de sistemas en base a piezas pequeñas y aisladas.

2. Por el otro, permite "extraer" dichas piezas y reutilizarlas en otros proyectos de sistemas lo que puede permitir crear a una comunidad de diseñadores el compartir sus diseños y reutilizarlos para construir sistemas más grandes.

En la parte superior se pueden ver los menús de la aplicación, de los cuales cabe destacar el menú "Modo", que se puede ver en la figura 6. Cuando la aplicación está en modo construcción permite insertar elementos, eliminarlos, configurar su funcionamiento y conectarlos unos a otros. Cuando la aplicación está en modo simulación solo permite conmutar la entrada de los elementos y observar su funcionamiento.

En el menú "Ver" se puede ver la vista XML del sistema representado en la página activa. El XML de un sistema determinado puede guardarse de forma separada para luego incorporar dicho sistema como subsistema de un sistema más grande. En las secciones de ejemplo se podrá ver el funcionamiento ampliado.

En la zona inferior de la página se puede activar o desactivar la construcción de vectores. Los vectores permiten construir conjuntos de elementos de forma más rápida que uno a uno así como modificar su posicionamiento relativo en pantalla. Se pueden insertar los elementos hasta de 10 en 10.

5. Un proyecto de ejemplo.

Para ilustrar el uso de SimProc se va a ver un ejemplo muy sencillo consistente en examinar las propiedades de un biestable.

Figura 7: Disposición de los elementos

Antes de empezar, se debe iniciar un nuevo proyecto usando el menú Archivo. Al elegir la opción "Nuevo" aparece un sistema vacío en el cual se podrán empezar a insertar los elementos.Para insertar primero el biestable, se debe pulsar encima del botón Biestable situado en el panel izquierdo. Una vez pulsado, el biestable está listo para ser insertado en el sistema mediante un click. Asi, pulsando una vez en la barra de herramientas y luego una vez en el sistema en blanco podremos ir añadiendo los elementos que sean necesarios. Repetir este proceso hasta que se obtenga un esquema como el que se muestra en la figura 7. Si se desea se puede encontrar este proyecto en el directorio `ejemplos`.

A continuación hay que conectar los elementos para indicar que la salida de unos elementos van conectados a la entrada de otros. Si se desea indicar que la salida de "Pulsador 1" va conectado a una entrada del biestable se debe hacer click encima del Pulsador y luego encima del biestable. Como el biestable tiene varias entradas se abrirá un cuadro de diálogo como el mostrado en la figura 8 para seleccionar la entrada a la que queremos conectar. Si se selecciona la entrada R el pulsador "Pulsador 1" quedará definitivamente conectado a la entrada R del biestable.

Repetir las conexiones conectando "Pulsador 2" a la entrada S y la salida "Salida Q" a "Lampara1" y "Salida Qneg" a "Lampara" hasta obtener un esquema como el mostrado en la figura 9

5.1. Simulación de un esquema

Una vez construido un sistema puede empezar a simularse cambiando el modo de funcionamiento en el menú "Modo". Al cambiar al modo "Simulación" no se podrán seguir haciendo conexiones ni cambios en el proyecto. Se puede empezar a interactuar con el circuito pulsando sobre los elementos.

En el proyecto inicial se puede probar a pulsar sobre los pulsadores, conectándolos y desconectándolos y observando como el biestable cambia de

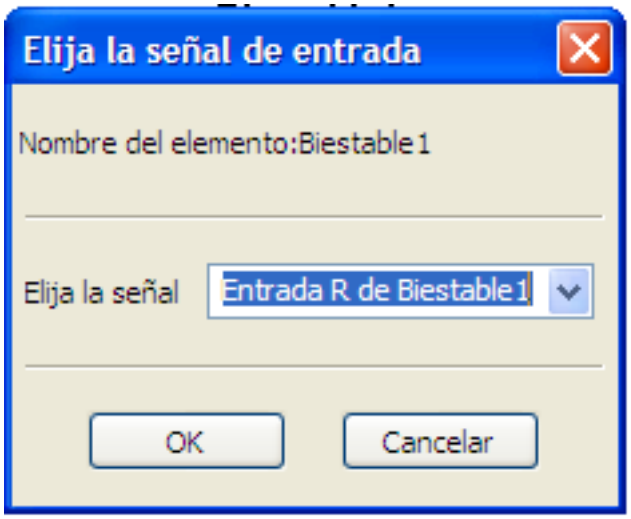

Figura 8: Cuadro de selección de la entrada

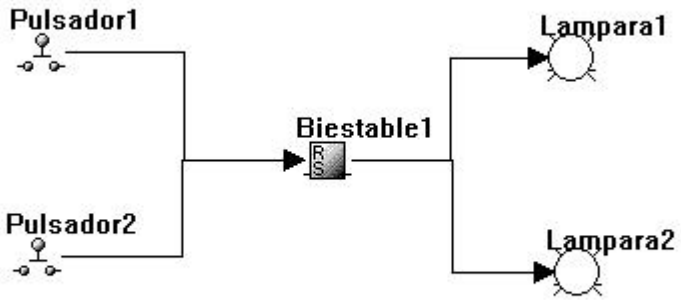

Figura 9: El primer proyecto listo para simular

estado y enciende las distintas lámparas en función de las entradas.

5.2. Errores de diseño

SimProc puede observar algunos errores en el diseño de sistemas, lo que se puede observar en el proyecto inicial dejando pulsados los dos pulsadores a la vez. Se observará un cuadro de diálogo como el mostrado en la figura 10 informando del error que se ha producido y se negará a ejecutar tal entrada.

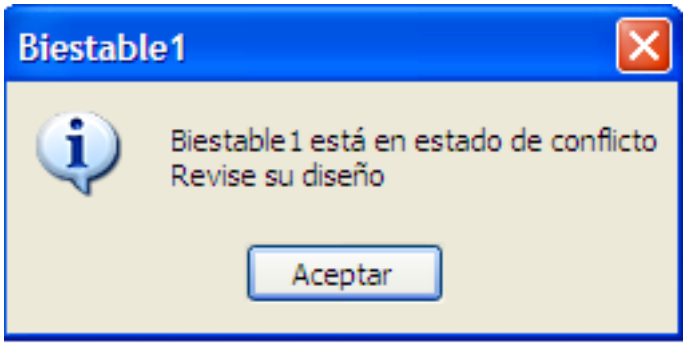

Figura 10: Un error en el diseño

5.3. Reconfiguración de los elementos

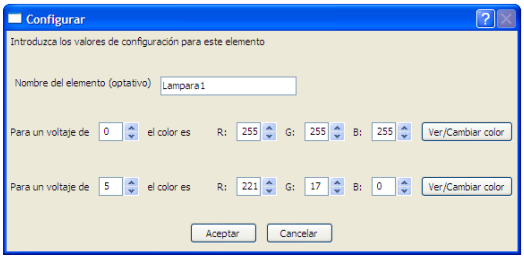

Figura 11: Un ejemplo de reconfiguración

Cuando se está en el modo construcción se puede modificar el comportamiento de los elementos usando el botón derecho del ratón. Si por ejemplo se pulsa encima de una Lámpara, se observará un cuadro de diálogo como el siguiente donde se permite cambiar los voltajes necesarios para cambiar de estado, los colores que muestra la lámpara o el nombre que tiene la misma. En la figura 11 se puede ver el resultado.

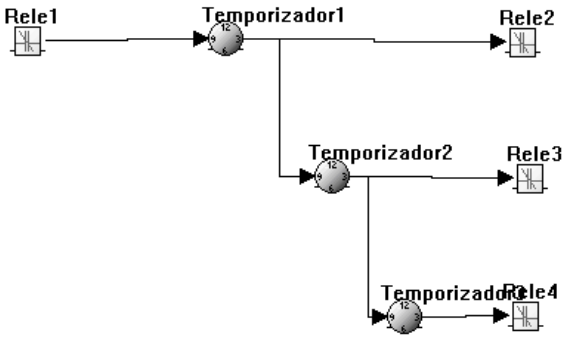

Figura 12: Un ejemplo de sistema temporizado

6. Un proyecto más grande

En esta sección ilustraremos algunas características más avanzadas de SimProc construyendo un sistema que encienda una serie de 3 contactos con un intervalo de tiempo entre ellos. Para ello usaremos varios temporizadores que se conectarán unos a otros de la forma que se muestra en la figura 12.

Cuando el temporizador 1 ha contado el intervalo prefijado (se puede configurar usando el botón derecho) activa el relé y el siguiente temporizador. Este efecto cascada hará conectar todos los contactos de forma ralentizada.

Si se inicia la simulación y se hace click en el relé disparador, se podrá observar como los relés se van activando a medida que los temporizadores alcanzan su tiempo límite.

7. Un sistema complejo

En ocasiones, los sistemas son demasiado grandes como para analizarlos de un solo vistazo y mucho menos diseñarlos de una sola vez. Al igual que en el software se pueden construir bibliotecas reutilizables, SimProc permite la construcción de sistemas que puedan tratarse como cajas negras. Utilizando el menú "Insertar->Subsistema" o el panel de la derecha se puede insertar un sistema vacío que se muestra como una solapa adicional en la ventana principal. Cuando se desee conectar un elemento con otro elemento que esté dentro de un sistema se puede hacer click sobre el icono del sistema que lo contiene y elegir el elemento con el cual se quiere interactuar. En la figura 13

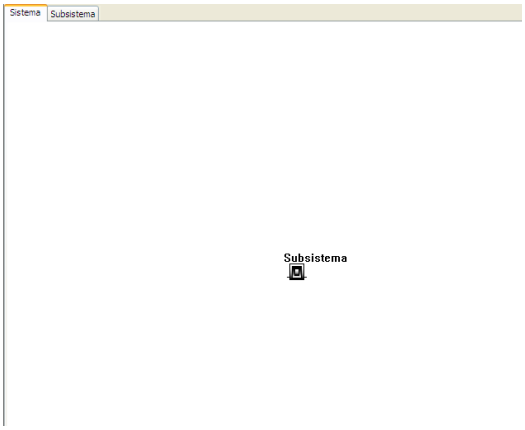

Figura 13: Un ejemplo de subsistema

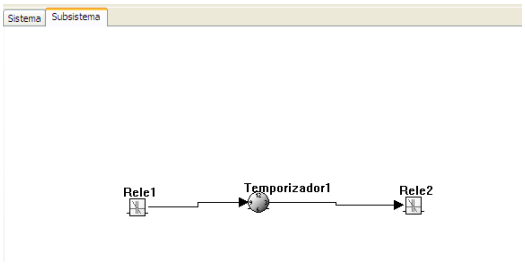

Figura 14: El subsistema por dentro

se puede ver un sistema dentro del cual se ha añadido un subsistema. Se puede seguir trabajando dentro del subsistema y conectarle luego las entradas que sean necesarias así como tomar de él las salidas que sean necesarias. En la figura 14 se puede ver que dentro del subsistema se ha añadido un sistema temporizado que retrasa la salida. Si se desea se puede volver a al sistema principal y conectar entradas hacia el Relé 1 y tomar la salida del Relé 2, tal y como se muestra en la figura 14.

8. Ejemplos adicionales

En el directorio `ejemplos` se pueden encontrar ejemplos adicionales de sistemas que se pueden abrir para examinar su funcionamiento así como reutilizar en proyectos más grandes.

9. Reutilización de los diseños.

Cualquier sistema ya diseñado se puede incorporar como parte de un sistema mayor utilizando el menú "Insertar->Subsistema desde archivo". Al abrir un sistema cualquiera se verá como dicho sistema se ha insertado dentro del diseño actual como un elemento más y con el cual se puede interactuar por medio de sus entradas y salidas.

Utilizando esta filosofía, es posible colaborar entre varios usuarios para construir sistemas partiendo solamente de la definición de entradas y salidas que cada subsistema necesita. Asimismo, cualquier usuario puede cambiar su diseño a posteriori y permitir su incorporación dentro de un sistema mayor sin necesidad de interferir con los demás usuarios.

10. Formato de archivo

SimProc almacena sus ficheros de proyecto en un vocabulario XML cuya gramática está definida en un esquema XML. Cualquier interesado en ampliar el programa deberá controlar que los nuevos elementos que se añadan, produzcan un XML válido según dicha gramática. El vocabulario XML define los tipos básicos de señales utilizados por el programa así como la sintaxis necesaria para la descripción de los elementos. Un elemento que desee almacenar su descripción en XML así como ser recuperado desde una descripción XML debe incorporar su sintaxis al fichero de esquema.

11. Errores

SimProc es un proyecto en desarrollo y como tal podría contener errores. Durante el desarrollo de la ejecución, la aplicación produce un fichero denominado `SimProc.log` en el cual la aplicación va tomando nota de los sucesos producidos. Si se detecta un error, se puede enviar esta traza al desarrollador así como el fichero de proyecto que dió problemas y de esta forma facilitar la corrección de errores.

En cualquier caso, al disponer del código fuente se puede examinar el código con ayuda de un depurador para intentar localizar y subsanar el error.