

UN ENTORNO DE MODELADO INTELIGENTE Y SIMULACIÓN DISTRIBUIDA DE PLANTAS DE PROCESO

Acebes L. (1), Alves R. (2), Merino A. (3) y de Prada C. (1)

(1) *Departamento de Ingeniería de Sistemas y Automática. Facultad de Ciencias. Universidad de Valladolid. C/ Doctor Merguellina s/n 47011, Valladolid, España.*

E-mail: {felipe,prada}@autom.uva.es

(2) *Department of Automation and Systems. Federal University of Santa Catarina. 88040-900 CTC-UFSC. Florianópolis-SC Brazil. E-mail: ralves@das.ufsc.br*

(3) *Centro de Tecnología Azucarera. Universidad de Valladolid. Edificio Alfonso VIII. 47011, Valladolid, España. E-mail: alejandro@cta.uva.es*

Resumen: Se describen un conjunto de aplicaciones informáticas que abordan diferentes aspectos del modelado y simulación de procesos continuos. Primero, se presenta un prototipo universitario SIMPD, que es un generador de código de simulación de sistemas de la industria de proceso y cuyo algoritmo básico trata de emular el modo de razonamiento de un experto en modelado cuando escribe un modelo de simulación. Se compara con otros enfoques y se analizan tanto las ventajas que presenta desde la perspectiva del usuario final como los inconvenientes para el programador que trate de aumentar el conjunto de sistemas modelables. Segundo, se describe tanto una arquitectura de simulación distribuida cuyas comunicaciones se basan en el estándar *de facto* OPC (*OLE for Process Control*) como el conjunto de herramientas informáticas desarrolladas para diseñar estos escenarios de simulación. Esta arquitectura se aplica a un proceso industrial, explicando el criterio usado para dividir el modelo de simulación global. Copyright © 2004 CEA-IFAC.

Palabras Clave: Modelado de sistemas continuos. Simulación distribuida.

1. INTRODUCCIÓN

Las técnicas de simulación de procesos continuos están bien establecidas, sin embargo eso no es completamente cierto para las técnicas de modelado. Existen diversos enfoques que abordan la problemática del modelado y de la reutilización de modelos, algunos de ellos son:

- Los lenguajes orientados a bloques como SIMULINK (Mathworks, 2004) o las interfaces gráficas de los lenguajes de simulación orientados a sentencias como GRAPHIC MODELLER (AEGis, 2004).
- Las técnicas de grafos de ligaduras conocidos como *bond graphs* (Cellier, 1991).
- Los lenguajes de modelado orientados a objetos (OOML) como Modelica (Modelica, 2004), implementado por DYMOLA (Dynasim, 2004) o MathModelica (MathCore Engineering, 2004), o Ecosim Language (EL) implementado por EcosimPro (EA International, 2004).
- Los lenguajes de modelado gráfico, donde los modelos dependen del contexto (Piera, 1993).
- Las herramientas basadas en técnicas de inteligencia artificial que abordan el problema de

la reutilización componiendo fragmentos de modelos y añadiendo conocimiento físico a la tarea de modelado. Entre estos enfoques se pueden destacar los trabajos relacionados con el modelado composicional (Nayak, 1995), la teoría de fenómenos híbridos de Woods (Woods, 1993) y los trabajos de Wasb0 (Wasb0, 1996) que plantean el modelado de unidades de proceso usando las técnicas de orientación a objetos y la topología interna de dichas unidades.

Desde la perspectiva de los autores, cada una de estas metodologías presenta inconvenientes desde el punto de vista del usuario (industrial) final.

En concreto los lenguajes orientados a bloques y las interfaces gráficas de los lenguajes de simulación orientados a sentencias no soportan la reutilización de modelos porque, entre otras razones, los bloques tienen la causalidad computacional ya asignada, la interconexión entre dichos bloques se hace a nivel de variables y no de conectores físicos, y la conexión de bloques puede hacer que aparezcan lazos algebraicos de difícil solución.

Los grafos de ligaduras si soportan la reutilización de modelos, pero su formulación resulta alejada de la formulación matemática de las ecuaciones del modelo matemático y compleja cuando deben manejarse muchos tipos de variables diferentes (como es el caso de la industria de proceso en la que

podemos encontrar caudales, temperaturas, entalpías, presiones, diversas concentraciones...).

Los OOML resultan la mejor alternativa para soportar de un modo eficiente la reutilización de modelos en el ámbito de la industria de proceso. Sin embargo su generalidad y planteamiento implican una serie de desventajas que dificultan su uso por personal no cualificado en el campo del modelado y simulación de sistemas continuos. Entre ellas podemos mencionar que:

- En ocasiones, durante el proceso de la asignación de la causalidad matemática del modelo, se requiere la ayuda del modelador.
- La descripción matemática de un mismo sistema con distintos grados de detalle se hace complicada, ya que el usuario debe seleccionar, para cada entidad física que componen el sistema global, uno de entre los diversos modelos que representan cada una de ellas. Esto implica que los múltiples modelos de una misma entidad física deban agruparse y clasificarse en una librería de modo que uno de ellos pueda seleccionarse en función de las suposiciones de modelado. En ocasiones la organización de los modelos en función de las diversas suposiciones no es fácil de realizar usando un mecanismo de herencia simple y mucho menos fácil es mantener esa librería de modelos, ya que el hecho de hacer diferentes suposiciones en ocasiones no implica la sustitución de una ecuación por otra, sino que puede implicar un cambio en la formulación del modelo.
- Por otro lado, conseguir que el modelo de simulación resultante de la descripción de un sistema usando un OOML sea numéricamente robusto y eficiente, por ejemplo evitando lazos algebraicos no lineales de gran tamaño, puede requerir que el usuario tenga que realizar ciertos “trucos” que implican un conocimiento detallado de los modelos componentes y cuya implementación se aleja de las especificaciones de la orientación a objetos.

Finalmente, los lenguajes de modelado gráfico y el uso de sistemas expertos son metodologías que aportan soluciones a algunos de los problemas enumerados, pero su uso y desarrollo se sitúan a un nivel académico y no industrial.

En este artículo se propone un enfoque diferente que trata de eliminar algunos de los problemas mencionados con los que se encuentra el usuario final de las herramientas de modelado. Estas ideas se implementaron en una herramienta informática (SIMPDP) que será descrita en el segundo epígrafe y que podríamos considerar un prototipo universitario a incluir dentro de las herramientas que aportan nuevas soluciones al modelado de sistemas.

Por otro lado, y con respecto a la ejecución del modelo de simulación, es bien conocido que cuando se requiere capacidad de ejecución en tiempo real del

mismo y este es de tal magnitud que no es posible su simulación en una sola máquina, es preciso dividir el modelo de simulación y realizar una ejecución distribuida en diferentes computadores.

Para realizar simuladores distribuidos existen algunos estándares, como HLA o CAPE-OPEN, que dan una serie de especificaciones sobre los componentes, de modo que estos puedan ser desarrollados y reutilizados por terceras personas fácilmente.

HLA (DoD, 2002), *High Level Architecture*, es un estándar que pertenece al *Defense Modeling and Simulation Office del Department of Defense (DoD)* de los EEUU, siendo aprobado como estándar abierto por la IEEE en septiembre de 2000 (IEEE Standar 1516). HLA usa sus propias librerías de comunicaciones y sincronización (RTI, *Runtime Infrastructure*). CAPE-OPEN (CAPE-OPEN, 1999), *Computer Aided Process Engineering Open Simulation Enviroment*) es un proyecto patrocinado por la UE que realiza las especificaciones de las interfaces de comunicación de los componentes del modelo para dos *middlewares*, OMG CORBA y Microsoft DCOM.

2. SIMPD: UN ENTORNO DE MODELADO DE PLANTAS DE PROCESO CONTINUO

Los autores han constatado que algunas de los problemas que presentan los entornos de modelado comerciales podrían evitarse si dichas herramientas emulasen el modo de razonamiento que sigue un experto cuando escribe un modelo de simulación (con la causalidad computacional ya resuelta). Lo cual requiere combinar el uso de la inteligencia artificial con las técnicas de modelado gráfico

Con esta premisa se desarrolló SIMPD (Sistema Inteligente de Modelado de Procesos Dinámicos), concebida como una herramienta de modelado gráfico de procesos de fácil uso para personas que no dominen las técnicas de modelado y simulación, pero que conozcan el proceso que quieren simular.

Aunque las ideas son aplicables a otro tipo de dominios, inicialmente se ha orientado hacia la industria de proceso azucarero (Acebes et al., 2001a). Siendo sus objetivos iniciales el diseño de estructuras de control y selección de estrategias de producción (Acebes et al., 1999) y la generación de código de simulación para un simulador de entrenamiento de personal de factorías azucareras.

Desde el punto de vista del usuario, el modo de modelar un sistema en SIMPD puede resumirse en cinco fases (*Figura 1*):

- **Conceptualización**, que se realiza de modo gráfico, seleccionando un conjunto de unidades de proceso y control, especificando sus características y conectándolas de un modo topológico hasta construir un diagrama P&I (*Figura 2*).

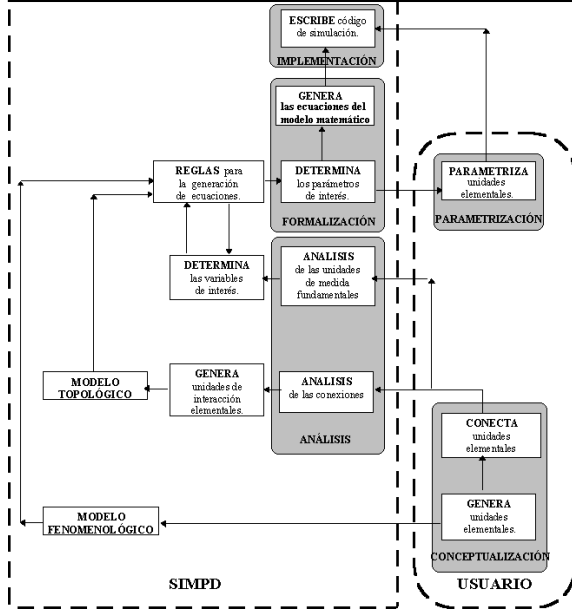


Figura 1. Esquema de funcionamiento de SIMPD

Esta forma de describir un sistema en base a la selección de componentes y a su interconexión topológica es similar a lo que ofrecen las interfaces gráficas de los OOML y resuelve el problema que presentan los lenguajes orientados a bloques en los que la conexión entre bloques que representan sistemas físicos puede implicar la realización de múltiples conexiones de variables asociadas a un único conector físico. Por ejemplo, imagínese que se desea conectar una válvula a una tubería situada en la base de un depósito, en esa conexión pueden estar implicadas múltiples variables (temperaturas, presiones, caudales máscicos, densidades, entalpías, ...) que aparecen de forma implícita al realizar la conexión física.

Por otro lado, la selección de las características de cada unidad de proceso y de los fenómenos a modelar especifica el modelo fenomenológico del sistema. Lo cual deriva de una forma implícita en la determinación de la granularidad del modelo y de las suposiciones de modelado. Esta especificación de fenómenos a modelar permite resolver de un modo elegante la selección de un modelo de una misma entidad física de entre todos los posibles modelos de la misma.

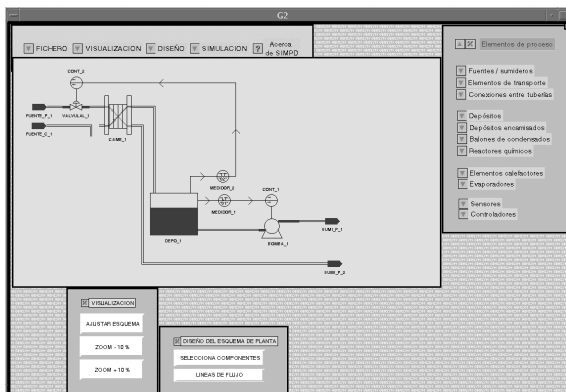


Figura 2. Descripción de un sistema en SIMPD

- **Análisis.** En base a la anterior descripción gráfica del sistema a modelar, se encuentran las interacciones entre las unidades componentes del sistema completo (tanto de las unidades de proceso, como de las de medida y control). Estas interacciones denominadas, respectivamente, líneas de flujo y de control constituyen el modelo topológico del sistema.

Este análisis también determina el conjunto inicial de variables de interés del modelo de simulación. Para expresar interés en una determinada variable debe situarse en el diagrama P&I el correspondiente elemento indicador.

- **Formalización.** Existe un algoritmo que, usando una librería de reglas de modelado para construir ecuaciones causales, genera el modelo de simulación dinámica del sistema. Cada una de las reglas de modelado están asociadas a una magnitud física y codifican como construir (escribir) una ecuación matemática causal, algebraica o diferencial, que permita calcular la magnitud física asociada a la regla.

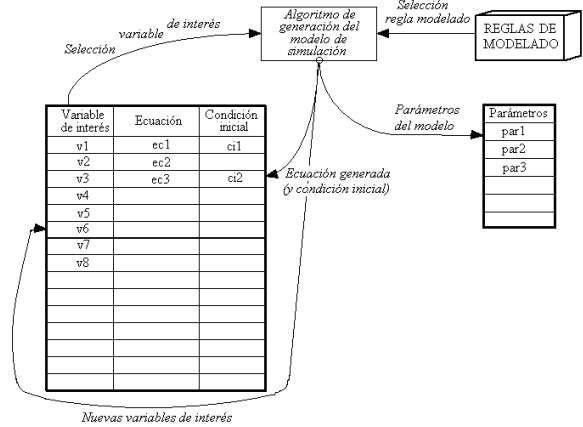


Figura 3. Generación del modelo de simulación

El algoritmo, *Figura 3*, usa como datos el conjunto inicial de variables de interés y los modelos topológicos y fenomenológicos del sistema descrito. La generación de la ecuación asociada a una variable de interés implica que deban generarse nuevas ecuaciones para las variables que contienen (que pasan a ser nuevas variables de interés), de modo que al finalizar el proceso de generación de código el modelo del sistema sea completo.

Esta forma de generar el modelo matemático evita los problemas asociados a la manipulación simbólica de ecuaciones, que conducen a modelos de una innecesaria complejidad numérica o a que el usuario tenga que descender al nivel de las ecuaciones para resolverlos.

- **Parametrización.** Para poder simular el modelo obtenido debe parametrizarse, es decir el usuario debe asignar manualmente el valor de los parámetros físicos (constantes) del sistema y del estado inicial del modelo. Por ejemplo en la *Figura 4* se observa como el usuario está

especificando la curva de carga de la bomba sin más que introducir los valores numéricos de la tabla que la definen.

- **Implementación.** Finalmente, las ecuaciones generadas se traducen a un lenguaje de simulación de sistemas continuos profesional.

En las primeras versiones de SIMPD sólo se generaba código de simulación para el lenguaje ACSL, *Advanced Continuous Simulation Language*, (AEgis, 2004). En versiones posteriores se han aumentado las capacidades de SIMPD de modo que adicionalmente pueda generar el código de simulación de acuerdo con la sintaxis de EL. Sin embargo, aunque el lenguaje EL soporta los paradigmas del modelado dinámico de sistemas continuos orientado a objetos, el código generado por SIMPD no hace uso de esas capacidades adicionales de EL sobre ACSL, ya que SIMPD generará un código de simulación monolítico (un único componente) y además con la causalidad computacional ya resuelta, de modo que el usuario no tenga que proponer ni condiciones de contorno, ni variables de ruptura de lazos algebraicos no lineales, ni tampoco resolver problemas de índice superior.

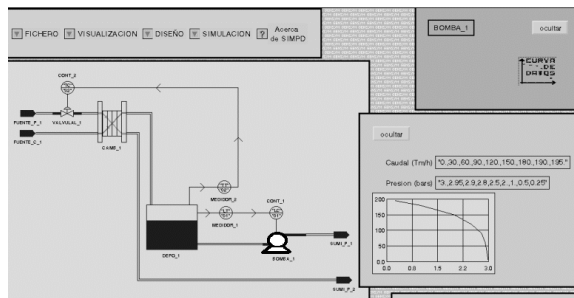


Figura 4. Parametrización de una bomba centrífuga

La ventaja de generar código EL, con respecto a generar código ACSL, es que la herramienta de simulación EcosimPro a partir de un modelo dinámico EL genera un modelo de simulación en forma de clase VC++ que puede ser incluido fácilmente en el desarrollo de cualquier aplicación informática en VC++. Utilizaremos esta característica de EcosimPro para realizar un conjunto de aplicaciones que permitan construir escenarios de simulación distribuida de plantas de proceso.

3. DESARROLLO DE APLICACIONES DE SIMULACIÓN DISTRIBUIDAS CON COMUNICACIONES BASADAS EN OPC

Para abordar la simulación distribuida con simulaciones desarrolladas con EcosimPro surge la posibilidad de aplicar algunas técnicas como las mencionadas en la introducción. Sin embargo, dado que el entorno de uso de las simulaciones desarrolladas van a ser aplicaciones relacionadas con el control de procesos, nuestra elección fue usar un estándar emergente llamado OPC, *OLE for Process Control*, (Chistrolm A., 1998). OPC está pensado para desarrollar sistemas abiertos en el entorno

Windows, basado en DCOM y orientado hacia las aplicaciones informáticas del ámbito del control de procesos y la industria de la manufactura.

La aplicación del estándar de comunicaciones OPC a las simulaciones generadas con EcosimPro, permite que estas actúen como aplicaciones del tipo servidor OPC y sean accedidas por una gran cantidad de aplicaciones comerciales. Estas aplicaciones actuarán como clientes OPC, como algunos sistemas de control distribuido, sistemas SCADAs u otras posibles aplicaciones de desarrollo propio con diversas funcionalidades, como la identificación, optimización y control de procesos (Acebes, 2001b).

Así, se ha realizado una aplicación informática (CreaOPC), cuya interfaz puede verse en la Figura 5, que a partir de un modelo de simulación en forma de clase VC++ desarrollado con EcosimPro permite generar una aplicación informática cuya ejecución sea la simulación del proceso en tiempo real (o acelerado) y con capacidad para comunicarse con otras aplicaciones a través del interfaz OPC.

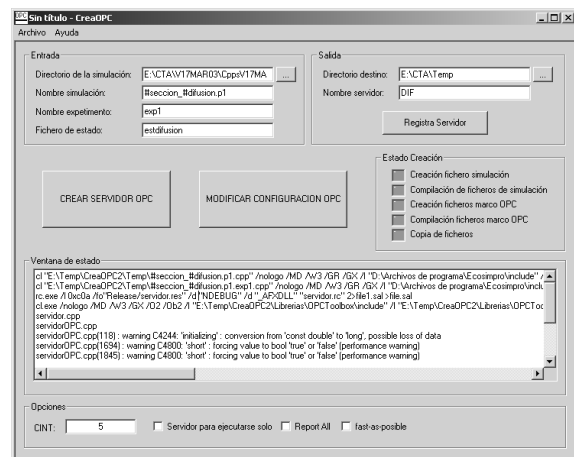


Figura 5. Interfaz gráfica de CreaOPC

Realmente, el ejecutable generado es una aplicación informática con tres partes. La primera, la simulación dinámica, la segunda, un hilo que controla el avance de la variable tiempo de la simulación y, la tercera, una capa de comunicaciones que le permite actuar como servidor OPC (Figura 6).

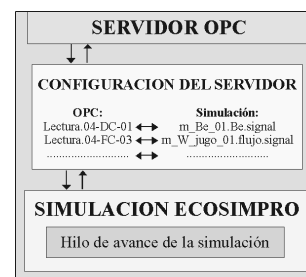


Figura 6. Estructura de un simulador con interfaz de comunicaciones OPC

Además, aprovechando que OPC está basado en DCOM y que permite que aplicaciones cliente accedan a servidores remotos ubicados en una red Windows NT, se puede dar un tratamiento

distribuido a las simulaciones generadas. Esto permite el reparto de carga de proceso cuando la carga computacional de las simulaciones supere las capacidades del computador en la que se ejecuten.

Así, una vez creados los servidores OPC correspondientes a las distintas partes del proceso que se quiere simular, estos se pueden ejecutar en un solo computador o en varios. Para la ejecución conjunta de todas las simulaciones es necesario que haya un intercambio de datos entre las distintas partes y una sincronización a fin de manejar el avance conjunto del tiempo.

El intercambio de datos y la sincronización se va a realizar a través de una aplicación de desarrollo propio que actúa como cliente de todos los servidores de las simulaciones, llamada UneSim. Esta aplicación, está programada de forma que cada intervalo de comunicación recoge los datos a intercambiar de los distintos servidores y los escribe en los servidores correspondientes. Todo esto se realiza de manera sincronizada, de forma que cuando termina de realizar el intercambio de datos permite a las simulaciones avanzar hasta el siguiente intervalo de comunicación (Figura 7).

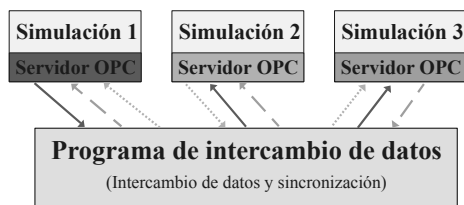


Figura 7. Escenario de simulación distribuida

La Figura 8 muestra la interfaz gráfica de UneSim en donde se puede observar la lista de pares de variables que deben intercambiar su valor y la gráfica de evolución del tiempo simulado frente al tiempo real, así como indicadores numéricos de dichos tiempos.

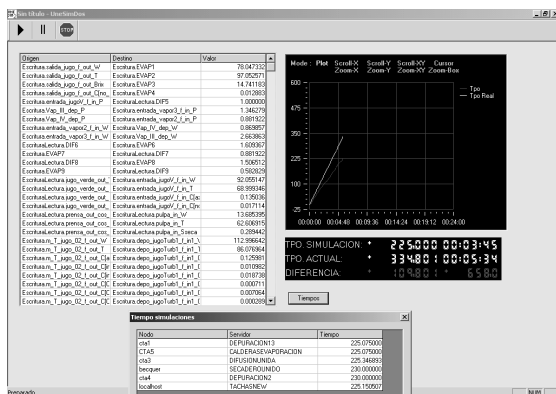


Figura 8. Interfaz gráfica de UneSim

Adicionalmente existe una aplicación denominada UnesimCnfg, que permite la configuración de UneSim de una forma sencilla. Especificando los servidores que van a conformar la simulación distribuida, la ubicación de los mismos y los pares de variables intercambiadas).

Esta arquitectura distribuida de simulaciones está siendo utilizada con éxito en un simulador de entrenamiento de operarios de sala de control de factorías azucareras (Figura 9). Funcionalmente se tiene una simulación distribuida en tiempo real que emula el comportamiento del proceso de fabricación del azúcar y su sistema de control y un sistema SCADA configurado tanto como consola de operario como de instructor (Prada et al., 2003).

En este simulador el instructor, desde su consola del SCADA, dirige la sesión de entrenamiento. Los operarios, desde sus consolas en el SCADA, tienen que responder, a los cambios y fallos provocados por el instructor en la simulación del proceso productivo, de un modo similar a como lo harían en la fábrica.

La misma funcionalidad del sistema hace que tenga que distribuirse en varios PCs (consolas de operario e instructor). Pero adicionalmente, el requerimiento de que el sistema deba funcionar en tiempo real y la complejidad del modelo matemático del proceso implica que la simulación por si misma tenga que distribuirse en varios PCs usando la metodología propuesta en este artículo.

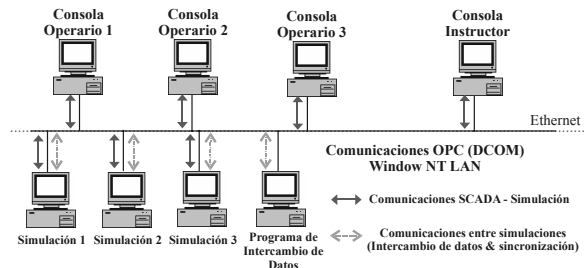


Figura 9. Arquitectura del simulador de entrenamiento

En concreto la planta se dividió en ocho simuladores, uno por sección del proceso productivo (Figura 10), excepto para la depuración que se dividió en dos, por tres razones:

- Primero, porque los simuladores de cada sección tienen entidad por si mismos y pueden ser utilizados de forma autónoma (reutilización de simuladores).
- Segundo, porque las variables a intercambiar entre secciones de proceso generalmente coinciden con magnitudes asociadas a los flujos de materia. Minimizándose el número de variables a intercambiar.
- Tercero, porque las magnitudes a intercambiar seleccionadas varían lentamente y permiten usar un intervalo de comunicación entre servidores de 5 segundos. Intervalo de comunicación lo suficientemente pequeño como para que no afecte significativamente a la resolución numérica de los modelos de simulación y suficientemente grande como para optimizar las comunicaciones.

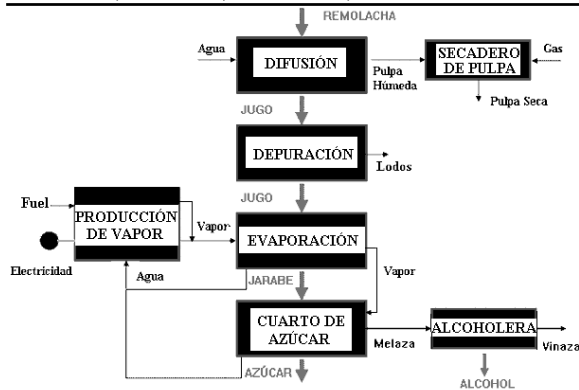


Figura 10. Esquemático del proceso productivo

Indicar que los simuladores de cada sección productiva no se han desarrollado con SIMPD sino con una librería de unidades de proceso azucareras desarrolladas por el Centro de Tecnología Azucarera (CTA) para la herramienta de simulación EcosimPro (Acebes et al., 2003). No obstante, el modelo de la sección de evaporación podría haberse desarrollado utilizando SIMPD, sección para la cual es operativo.

4. CONCLUSIONES

En este trabajo se ha presentado una metodología que permite automatizar y facilitar el desarrollo de escenarios de simulación distribuida de plantas de proceso. El desarrollo de esta metodología ha dado lugar a diversas aplicaciones informáticas: SIMPD, CreaOPC y UneSim (y UnesimCnfg).

SIMPD, es un prototipo universitario, no una herramienta comercial, que permite generar código de simulación dinámico (en forma de código ACSL ó EL) que se ha usado en el CTA para estudiar en simulación diversas estrategias de control del proceso de evaporación.

Las ventajas de SIMPD es que, primero, el modelo se describe de un modo gráfico, construyendo un diagrama P&I, seleccionando unidades de proceso y conectándolas. Segundo, la selección del grado de detalle del modelo matemático se realiza de un modo automático ya que el modelador especifica los fenómenos presentes en cada entidad física durante su selección. Tercero, no se requiere manipulación simbólica de las ecuaciones del modelo matemático resultante, ya que el modelo de simulación se genera directamente a partir de la base de conocimiento que contiene las reglas de modelado.

Las mayores desventajas de SIMPD son que, primero, la librería de entidades físicas es incompleta y la inclusión de la capacidad de modelado de una nueva entidad requiere la generación de una nueva clase de entidad y su correspondiente menú de dialogo para la especificación de características y fenómenos físicos presentes. Segundo, la inclusión de nuevas entidades puede requerir la inclusión de la descripción de nuevos fenómenos (reglas de modelado), que deben organizarse de acuerdo, y sin conflictos, con los ya existentes.

Actualmente la inclusión de nuevos modelos componentes resulta muy compleja y requiere del conocimiento del entorno de programación G2, en el que se ha desarrollado SIMPD, y de la estructura interna de la programación de SIMPD. Esto evidentemente dificulta el desarrollo de nuevas librerías de componentes seleccionables, haciéndose necesario el desarrollo de una aplicación informática que simplifique esta tarea.

Por otro lado es también necesario estudiar el modelado de sistemas híbridos complejos en SIMPD.

Con respecto a la simulación distribuida, se ha probado con éxito el uso de OPC como soporte para las comunicaciones en un simulador distribuido de un proceso industrial y se han desarrollado herramientas informáticas, a partir de modelos EL, permiten la creación (CreaOPC), configuración (UnesimCnfg) y ejecución (UneSim) de este tipo de simuladores.

El uso de OPC tiene muchas ventajas, la principal es que ha sido adoptado como estándar *de facto* de comunicaciones industriales por muchas compañías del sector del control de procesos. Esto permite que las simulaciones desarrolladas con la metodología propuesta, dispongan de un interfaz de comunicaciones que permitan su conexión a muchos de los sistemas de control y SCADA industriales.

Por otro lado, con respecto a la arquitectura distribuida de simulación debe estudiarse tanto la eficiencia de las comunicaciones al utilizar el estándar OPC, como las limitaciones que impone sobre el intervalo de comunicación cuando se quiere lograr que todo el sistema funcione en tiempo real.

Agradecimientos

Los autores quieren agradecer el soporte financiero de la empresa Ebro Agrícolas así como a la Junta de Castilla y León por medio del proyecto “*Desarrollo de un entorno de modelado inteligente y simulación distribuida de plantas de proceso*”.

REFERENCIAS

- Acebes L.F & Prada C. de (1999). Process and control system design using dynamic simulation. *CITS Proceedings 1999*. Pp. 73-81. Editorial Bartens.
- Acebes L.F & Prada C. de. (2001a). Rule Based Approach To First Principles Modelling In The Sugar Industry. *Advanced Control of Chemical Processes 2000* (Biegler L., Brambilla A. & Scali C. (Ed)), vol. 1, pp. 335-340. Editorial Pergamon Press.
- Acebes L.F & Zamarreño J.M. (2001b). Diseño de un SCADA usando un simulador OPC. *Automática e Instrumentación*. No. 316, pp. 76-80. 2001.
- Acebes L.F., Prada C. de, Alvés R., Merino A., Pelayo S., García A., Rueda A. & Gutiérrez G. (2003) Development Tools for Full Scale Simulators of Sugar Factories. *CITS Proceedings 2003*. Pp 131-138. Editorial Bartens.

- Aegis (2004). <http://www.acslsim.com>.
- CAPE-OPEN (1999). *Next Generation Computer-Aided Process Engineering Open Simulation Environment: Public Synthesis & Roadmap*.
- Cellier, F.E (1991). *Continuous System Modeling*. Editorial Springer-Verlag, New-York. Capítulo 7: *Bond Graph Modeling*, pp.251-296.
- Chistrolm A. (1998). *OPC Data Access 2.0 Technical Overview*. OLE for Process Control and Factory Automation. Intellution, Inc.
- DoD (2002). Department of Defense. Defense Modeling and Simulation Office. *High Level Architecture. Run-Time Infrastructure. RTI 1.3-Next Generation Programmer's Guide Version 5*.
- Dynasim (2004). <http://www.dynasim.se>.
- EA Internacional (2004). <http://www.ecosimpro.com>
- Mathworks (2004). <http://www.mathworks.com>.
- MathCore Engineering (2004). <http://www.mathcore.com>.
- Modelica (2004). <http://www.modelica.org>.
- Nayak, P.P (1995). *Automated Modelling of Physical Systems*. Lectures notes in Artificial Intelligence 1003. Editorial Springer-Verlag Berlin.
- Piera, M.A. (1993). *PMT: Un Entorno de Modelado en la Industria de Procesos*. Ph.D. Universidad Autónoma de Barcelona.
- Prada, C de, Merino A., Pelayo S., Acebes F. & Alves R (2003). *A Simulator of Sugar Factories for Operator Training*. II Int. Workshop on Mathematical And Computing Techniques For Agro-Food Technologies, MACSI-Net. Barcelona, España. Noviembre 27-28, 2003.
- Wasb0, S.O (1996). *Ferromanganese Furnace Modelling using object-oriented principles*. Ph. D. Department of Engineering Cybernetics. Norwegian University of Science and Technology. N-7034 Trondheim, Norway.
- Woods, E.A (1993). *The Hybrid Phenomena Theory*. Ph. D.