

Bases de diseño de un Nodo Virtual de Procesos.

Y. Gámez Batista¹, V. Moreno Vega², Y. Martínez Márquez³

1 Universidad de las Ciencias informáticas, Cuba, km 5 ½ carretera San Antonio de los Baños, Rpto. Torrens, Boyero, Habana (e-mail: yaliceg@uci.cu)

2 Instituto Superior Politécnico José Antonio Echeverría, Cuba. (e-mail: valery@electronica.cujae.edu.cu)

3 Universidad de las Ciencias informáticas, Cuba, km 5 ½ carretera San Antonio de los Baños, Rpto Torrens, Boyero, Habana. (e-mail: yoanm@uci.cu)

RESUMEN

En educación, los contenidos teóricos de las diferentes asignaturas son impartidos en clase pero esto no garantiza que los estudiantes adquieran las habilidades para hacer frente a los problemas reales. Estos contenidos se complementan con laboratorios y actividades prácticas orientados al desarrollo de habilidades prácticas. Desde hace unos años estas habilidades se refuerzan con la creación de laboratorios virtuales de diversa índole donde las nuevas tecnologías se ponen en función de ampliar las capacidades de simulación y prueba en entornos cercanos a la realidad. En este trabajo se aborda el diseño de un Nodo Virtual en el cual se implementan los modelos de distintos procesos, agrupados por tipos que dan la posibilidad de realizar simulaciones con los mismos o probar aplicaciones en tiempo real.

Palabras claves: Control Automático, Nodos Virtuales, Procesos Industriales, Simulación.

Design Bases for a Virtual Process Node.

ABSTRACT

In education, the theoretical contents of the subjects are taught by teachers in lessons, but this doesn't guarantee that the learners acquire the abilities they need to face real problems. These theoretical aspects are complemented with several practical activities and laboratories always oriented to develop of student's practical abilities. Since several years ago these abilities are reinforced in virtual laboratories in which the new technologies are used to wide up the tests and simulations capabilities in environments close to reality. This paper deals with the design of a virtual node, where they will be implemented the models of different processes, they are going to be group in classifications allowing the possibility to make simulations with them or testing applications in real time.

Key words: Automatic Control, Industrial Process, Simulating, Virtual Node.

INTRODUCCIÓN

Es indudable la importancia adquirida por las nuevas tecnologías de la información y comunicaciones (TIC) en todos los órdenes de la sociedad y, en particular, en el ámbito de la Educación Superior donde han introducido nuevas formas de trabajo, relación e, incluso, cambios en los métodos pedagógicos con los que se superan los métodos tradicionales de difusión de la documentación por parte del profesor [1].

El objetivo, a la hora de aplicar las TIC en la docencia, consta de [1], [2]:

- Proporcionar un escenario educativo nuevo caracterizado por la representación virtual del proceso de enseñanza (modelo centralizado en la enseñanza remota).
- Modelo que complementa la docencia presencial, sobre todo, como apoyo para el cumplimiento de las directrices marcadas por la Educación Superior. En la actualidad, a nivel mundial, la universidad presencial se encuentra con que cada vez más, sus estudiantes poseen sus propios computadores (en casa o portátiles) con posibilidad de conexión en red (wireless, LAN o conexión residencial cada vez con mayor ancho de banda) y con la esperanza de que se conviertan en una herramienta útil en su preparación universitaria a todos los niveles, incluso en la docencia práctica.

Como señalan diversos autores [3], [4], ya es un componente de excelencia universitaria el disponer de una infraestructura tecnológica que permita transformar los equipos propietarios de los estudiantes en potenciales clientes de laboratorios y aulas informáticas (incluso en aulas de teoría) que puedan utilizar los recursos hardware y software de la Universidad conectándose desde cualquier punto del campus o, en general, desde Internet.

Si, por otra parte, se tiene en cuenta que la formación de los alumnos universitarios en TIC ha aumentado progresivamente en los últimos años [5] se justifica sobradamente el apoyo que la mayor parte de las universidades está dando a la docencia virtual por Internet [6] en sus distintos modelos.

Existen múltiples universidades que utilizan la red y sus servicios para proporcionar información (directorio de profesores, alumnos y personal de administración y servicios), tareas de administración universitaria, calificaciones, etc. Sin embargo, estas acciones, si bien sirven de apoyo a la docencia, no utilizan las capacidades de las TIC para proporcionar un nuevo espacio para desarrollar actividades educativas [1].

La expansión de la disponibilidad del material educativo implica el desarrollo de recursos que puedan estar online — presentaciones de los profesores, materiales de investigación, datos, herramientas de análisis, programas multimedia y otras herramientas basadas en web— y accesibles desde todos los lugares de estudio incluyendo las propias redes inalámbricas desde cualquier punto del campus [7].

Sin embargo, un aspecto importante que se encuentran los alumnos durante su formación práctica, sobre todo en las titulaciones técnicas y experimentales, es la falta de disponibilidad y coste del software y el hardware necesario para realizarlas [8].

En el Instituto Politécnico José Antonio Echeverría (CUJAE), Facultad de Ingeniería Eléctrica, se estudia la especialidad de ingeniería Automática. En esta carrera se imparten una serie de contenidos que proveen a los estudiantes de toda la teoría necesaria para ejercer como profesionales de la rama. Sin embargo, no siempre estos conocimientos proveen a los estudiantes de las herramientas prácticas para enfrentarse a una situación real. Esto está muy asociado a la forma tradicional en la que se imparten estos tópicos y es entonces que se hace necesario el uso de las TIC.

Para dar solución a esta problemática se diseña un Nodo Virtual de Procesos que permita ejecutar diferentes instancias del software en un único nodo (nodo físico) y con cada instancia del nodo trabajando en un entorno de ejecución independiente (nodo virtual). Esta filosofía de trabajo permite la simulación simultánea de diferentes procesos concurrentes sin que interfieran unos con otros.

1. Nodos virtuales.

Durante el diseño e implementación de lazos de control es esencial analizar el efecto que tiene la acción de control sobre el proceso. Mientras que el análisis matemático y la simulación son comúnmente usados en las primeras etapas de diseño, la medición se usa para comprobar los resultados teóricos tan pronto como la implementación sea posible. Estas mediciones

son consideradas problemáticas por dos razones. Primero, en entornos reales, por las graves consecuencias que acarrearía cualquier error en el diseño e implementación del lazo de control. Segundo, los requerimientos de recursos no permiten mediciones de una gama amplia de procesos. Es por esto que existe una gran demanda de entornos de redes virtuales que pueden ser parametrizadas para reproducir un proceso real o ficticio.

Un nodo virtual de procesos es un software que permitirá implementar los modelos de diferentes procesos para su simulación o para probar aplicaciones en tiempo real. Será necesario que varios procesos se encuentren activos de manera simultánea, lo que requerirá de gran cantidad de nodos y computadoras que no están disponibles generalmente.

Sin embargo, un número de aplicaciones dirigidas a dispositivos de pocos recursos, sólo necesitan una fracción de los recursos que un nodo de prueba puede proveer. Además, se pueden ejecutar varios procesos en un único nodo de prueba (nodo físico). Cada proceso debe proveer un entorno de ejecución separado (nodo virtual).

¿Qué ventaja aporta un sistema como el nodo virtual de procesos frente a los laboratorios virtuales y remotos con una simulación distribuida? En el caso de esta aplicación con fines docentes, el hecho de tener las simulaciones en el servidor permite que:

- Sólo sea necesaria una computadora con altas prestaciones hardware para el servidor, y que las aplicaciones clientes puedan ejecutarse en cualquier computadora.
- Puedan conectarse usuarios, como observadores, a simulaciones previamente configuradas por otro usuario. De esta manera se propicia el trabajo en equipo y con ello un mejor aprovechamiento del ejercicio.

En general, la virtualización de nodos provee una vía de regular el acceso a recursos de hardware exclusivos de un determinado número de consumidores. En este caso los consumidores son los entornos de ejecución para cada proceso, los cuales están sujetos a las propiedades de la simulación [9]. De ahí se derivan los siguientes requerimientos para la virtualización de los nodos:

- El parámetro más importante es minimizar los gastos de virtualización del nodo para preservar los recursos para el proceso en ejecución.
- Cada entorno de ejecución introducido por la virtualización debe ser tan transparente como sea posible para los restantes. Esto es importante para que la medición de la implementación no sufra modificaciones en comparación con la real.

2. Antecedentes.

Las herramientas interactivas constituyen un elemento muy potente que permite reflejar la componente visual subyacente al concepto o problema en estudio bajo la abstracción de

conceptos matemáticos o de otro tipo, así como estimular la intuición de los estudiantes. De esta forma un alumno puede trabajar sobre un problema de forma gráfica y observar cómo el cambio en un determinado elemento se ve reflejado de forma inmediata en el resto, cómo si estuviera ante el proceso real [10].

Para la simulación de redes, en la Universidad de Stuttgart en Alemania, instituto especializado en sistemas distribuidos y paralelos, se creó la aplicación “Network Emulation Testbed” [9]. Esta aplicación va dirigida a simular un entorno de redes configurable que permita reproducir un escenario real de cientos de nodos en comunicación. De esta manera posibilita medir de manera comparativa el comportamiento de una aplicación en diferentes entornos de redes o de varias aplicaciones en un mismo entorno de red.

La principal limitación de esta aplicación está en que es diseñada para simular redes por lo que no es posible utilizarla para la simulación de procesos que tienen una dinámica un tanto más compleja y variada.

En la Universidad de Nottingham Trent, se creó un software para la simulación de sistemas de agua [16]. Surge por la necesidad que tenían en la industria del agua de adquirir y almacenar datos de estaciones remotas para la inspección ingenieril. La integración del sistema de supervisión con el software de simulación, para lograr un sistema capaz de tomar decisiones en tiempo real, conllevó a un incremento de los requerimientos computacionales para los algoritmos de simulación con un respectivo aumento de tamaño de la red física. Una solución clásica fue dividir la red en subredes para resolver las subredes de manera aislada y de esta manera coordinar las soluciones de los subsistemas para encontrar la solución del sistema completo.

Esta aplicación no es la solución pues no está concebida para simular procesos independientes, sino que simula subprocesos de un sistema específico, para luego integrarlo en el proceso general.

En el laboratorio Nacional de Lawrence Berkeley en EUA, se enfrentaban al problema de tratar las condiciones límites en los pozos de petróleo en el momento de formular y codificar un simulador numérico multifase de la reserva. Esto se debe a la complejidad de las ecuaciones diferenciales que gobiernan el flujo de la superficie que son una mezcla de tipo hiperbólica-parabólica que provocan problemas de convergencia computacional [17]. Por esta razón se utilizó el método de nodos virtuales donde a cada capa se le asignó un nodo virtual cuyo tratamiento está dado en dependencia de las características de la capa a la que represente para los cálculos del flujo. La solución en el pozo se obtendrá de resolver las ecuaciones del balance de masa para el nodo del pozo. De la misma manera que el anterior este simulador fue diseñado para el estudio de las diferentes capas de un pozo petrolífero para luego integrarlo en el modelo del pozo.

En la Universidad de Stanford se creó un algoritmo de nodos virtuales para el trabajo con imágenes en tres dimensiones

[18]. Un elemento es fragmentado para crear varias réplicas del elemento y se le asigna una porción real del material a cada réplica. De esto resultan elementos que contienen material real y regiones vacías. El material faltante está contenido en otra u otras copias. El algoritmo de nodo virtual determina automáticamente el número de réplicas así como la asignación de material para cada una. Provee además los grados de libertad requeridos para simular el material parcial o completamente fragmentado en una imagen consistente con la geometría. Aprovecha las posibilidades de la simulación de una geometría compleja con una simple mezcla. Este simulador tampoco satisface las necesidades planteadas.

En Japón se creó una herramienta “StarBED” para dar solución al vacío que existe entre internet y los entornos para experimentación, atendiendo a los aspectos de escala, complejidad y realidad [19]. Es una aplicación de pruebas basada en lotes de nodos que tiene como objetivos construir entornos de experimentaciones reales, complejos y de gran escala. Ya ha sido probada con efectividad en diferentes experimentos como son:

- Test de rendimiento de switches L2 con tráfico multiemitido.
- Observación del comportamiento del TCP.
- Comparación entre el comportamiento entre routers de hardware manufacturado por vendedores populares y routers de softwares en código abierto ejecutándose en una PC.
- Simulación de redes inalámbricas en redes cableadas.
- Simulación de redes de teléfonos celulares G4.

Estas herramientas fueron creadas para evitar las influencias de los servicios críticos de internet en las aplicaciones de simulación. Se utilizan para la simulación de diferentes tipologías de redes y no están concebidas para la simulación de procesos. Por esta razón no satisface los requerimientos de la aplicación planteada.

En la actualidad existen varios laboratorios virtuales y remotos que permiten realizar simulaciones de procesos industriales. Dentro de ellos se pueden mencionar Cockpit [11], la red de laboratorios Automatl@b [12], eElab [13] montadas sobre el entorno de colaboración eMersion [14] y el RTAI-lab [15], etc.

Estos laboratorios se caracterizan de manera general por permitir la simulación virtual o remota. En las aplicaciones servidor utilizan como software Matlab i/o Labview y en las aplicaciones clientes Easy Java Simulation. Matlab y Labview son propietarios lo que constituye un gran inconveniente para la Universidad. Por otro lado las simulaciones virtuales se realizan en las aplicaciones clientes lo que limita el trabajo grupal, como se señaló anteriormente, y es uno de los requerimientos de la aplicación. En el caso de RTAI-lab, es una aplicación libre para tiempo real pero, lo mismo que las anteriores, sólo permite la conexión de un usuario en cada simulación o tarea. Todos estos laboratorios, además, tienen la

limitación de que no permiten conectar controladores físicos a los procesos simulados que es una de las principales exigencias de la aplicación deseada.

De manera general se puede concluir que en la actualidad los nodos virtuales son potencialmente usados para trabajar en aplicaciones diseñadas para la simulación de redes, para la simulación de procesos que por su complejidad no pueden ser realizadas por las herramientas tradicionales y para el tratamiento de imágenes, pero no se ha identificado una herramienta informática que funcione como un nodo virtual de procesos, sobre todo, satisfaciendo los requisitos de diseño que den respuesta a las necesidades de la problemática planteada en la Introducción.

La realidad es que no existe una herramienta de libre distribución que, de manera remota, permita a los estudiantes interactuar, de manera individual o por equipos, con diferentes procesos configurados por ellos o por su profesor, y con los que puedan probar sus estrategias de control (ya sea virtual como físicamente), y poner en práctica los conocimientos adquiridos en clase.

A continuación se describen los requisitos y funcionalidades del nodo virtual de procesos que se propone en este artículo, así como aspectos asociados a la metodología de diseño empleada, selección de la herramienta y lenguaje de programación y de base de datos. Finalmente, se presenta un esquema funcional de trabajo del Nodo Virtual de Procesos así como un ejemplo sencillo que ilustra el funcionamiento de acuerdo a los diferentes actores.

MATERIALES Y MÉTODOS

1. Requisitos del nodo virtual de procesos.

La función principal del NVP es implementar los modelos de diferentes procesos o plantas de diverso tipo de la manera más real posible. En lugar de contar con una maqueta del proceso, o de un proceso real a escala reducida, se puede tener un nodo que se pueda acceder desde un laboratorio y que sirva para emular al proceso real, permitiendo especificar valores a la entrada del proceso y leyendo a su salida, de la misma forma que se podría hacer en una instalación física. De esta manera, se simularían lazos de control conectando el NVP a esquemas de control.

Es necesario entonces, cuando se implemente el proceso en el NVP, establecer la forma, metodología en que debe hacerse, así como las funcionalidades que tendrá el nodo para una vez implementado un proceso poder utilizarlo.

El NVP simula un proceso real mediante un modelo (modelo que debe describirse de cierta forma), por tanto, debe ser capaz de para un valor de entrada devolver la salida real. Una vez que el NVP esté activo, los usuarios podrán abrir un proceso, ejecutarlo, modificar sus parámetros, etc. Estos usuarios podrán ser simples observadores (hacen uso del NVP pero no lo modifican) o administradores.

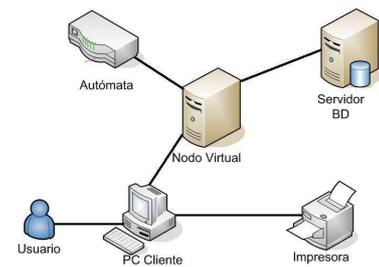


Figura 1. Esquema de la aplicación.

La formalización de estas necesidades se establece mediante la descripción de requisitos. A continuación se presentan algunos:

- Esta aplicación permitirá interactuar con los modelos de diferentes procesos, ya sea a través de simulaciones como probando aplicaciones en tiempo real. Esto se aprecia en la figura 1.
- Estos modelos se agruparán por tipos y serán accedidos por aplicaciones clientes que se conectarán al nodo mediante un protocolo establecido. El nodo virtual permitirá la conexión en red tanto desde una computadora personal como desde un autómata.
- El número de procesos activos simultáneamente es limitado, así como el número de clientes por proceso. Existen además niveles de prioridades para la cantidad de procesos por tipo (mecánicos, eléctricos, hidráulicos, térmicos, etc.) que se pueden activar. Las prioridades son necesarias para garantizar un nivel de prestaciones adecuado en el nodo en cualquier momento y prever situaciones de saturación de usuarios, recargas, etc.
- En la figura 2 se observa que los clientes tendrán diferentes niveles de privilegio: cliente común, cliente maestro y cliente administrador. Estos niveles se escogen al iniciar el trabajo desde una aplicación remota. Cada proceso activo tendrá un cliente maestro y varios clientes comunes.



Figura 2. Actores de la aplicación.

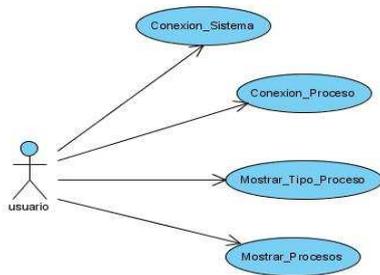


Figura 3. Esquema de Casos de uso del cliente común.

- Como se observa en la figura 3 los clientes comunes son aquellos que trabajan con un proceso ya activado y el número de ellos por proceso será definido por el sistema. Los parámetros del proceso serán establecidos por el cliente maestro, y los clientes comunes trabajarán con dichos parámetros. Esto se aprecia en la figura 4. Estos clientes pueden trabajar como espectadores del proceso o podrán interactuar con una instancia del proceso al que se hayan suscrito.

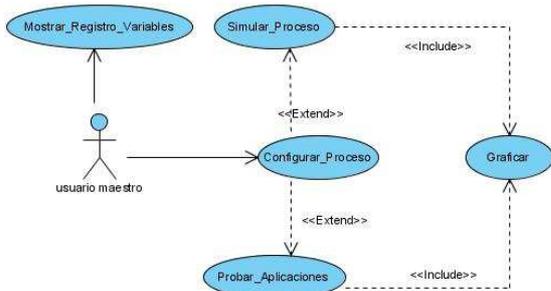


Figura 4. Esquema de Casos de Uso del cliente maestro.

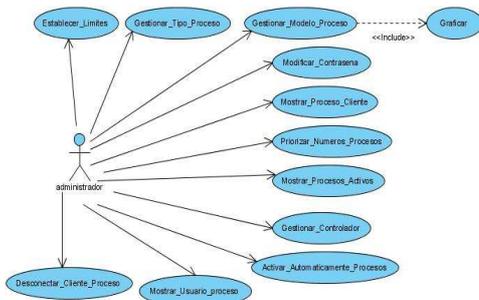


Figura 5. Esquema de Casos de Uso cliente administrador.

- El cliente administrador no actúa sobre los modelos sino que tiene acceso a otro tipo de información y ejecuta otras tareas vinculadas a la gestión del sistema. Esta categoría implica que el cliente administrador debe poseer un nombre de usuario y contraseña para poder registrarse como tal. Para más detalles observar la figura 5.

1.1 Un ejemplo de funcionamiento de los roles de cliente, cliente maestro y administrador.

Supongamos que se quiere introducir un modelo nuevo al NVP. El modelo, desde el punto de vista del NVP, es una representación del proceso real, que en su forma más general, puede ser un sistema de ecuaciones (posiblemente diferenciales) con un vector de entradas definido, un vector de salidas y una serie de parámetros.

De acuerdo a los requisitos el responsable de registrar los nuevos modelos es el administrador, quien a través de una interfaz amigable introducirá el sistema de ecuaciones así como definirá las variables de entrada, las de salida y los parámetros del modelo. Este modelo, además, será ubicado como un proceso inactivo en uno de los grupos definidos, según el tipo dentro del que clasifique. Supongamos, para los fines de esta explicación, que el modelo es $f(x) = mx+n$.

Un usuario al entrar al sistema se identificará, y tendrá la posibilidad de listar los procesos existentes en el NVP agrupados por tipo, y podrá identificar los activos y los inactivos, y conectarse a uno de ellos. Una vez conectado se creará un registro de todas sus acciones dentro del sistema. En este caso, al listar los procesos inactivos aparecerá el inscrito por el administrador, o sea, $f(x) = mx+n$. Si el usuario quiere activar el proceso, deberá entonces definir los valores de los parámetros m y n , y cuando lo comience a ejecutar se convertirá en un proceso activo con parámetros definidos (por ejemplo para $m=4$, $n=-1$ el proceso sería $f(x)=4x-1$). En este caso el usuario se convierte en cliente maestro del proceso $f(x)=4x-1$.

Si otro usuario desea conectarse, aparecerá como proceso activo el anterior. En caso de conectarse a un proceso ya activo, es decir un proceso previamente configurado, se convertiría en un cliente común, y recibirá todos los datos de la simulación de dicho proceso. Se comporta como un observador del mismo y no puede ejecutar ninguna acción sobre él. Si por el contrario, quiere trabajar con el mismo modelo, pero con otros parámetros, podrá definir un nuevo proceso activo, por ejemplo, para $m=5$ y $n=2$. Pasará a ser también cliente maestro y se tendrá en el NVP en ese momento dos procesos activos, con parámetros diferentes.

Aunque el ejemplo se ha simplificado, lo mismo ocurre con el caso del modelo de una caldera, una torre de destilación, un sistema de tanques interactuantes, etc.

En caso de que seleccionara un proceso inactivo tendría que configurar los parámetros del mismo y se convierte en su usuario maestro. Esto le da la posibilidad de manipular el proceso a conveniencia. También pudiera darse el caso de que deseara conectarse a un proceso ya activo pero con otra configuración. El sistema le da la posibilidad de crear una nueva instancia del proceso y actuaría como cliente maestro de la misma.

2. Metodología de diseño y lenguaje de programación.

La metodología de diseño utilizada fue RUP. RUP es un proceso de desarrollo de software, definido como un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software, está basado en componentes y utiliza UML (Lenguaje de Modelado Unificado (Unified Modeling Language)) para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

Está dirigido por casos de uso (forma en como un actor opera con el sistema en desarrollo), porque con éstos se especifican las funcionalidades que el sistema proporciona al usuario. Los casos de uso representan los requisitos funcionales y fuerzan a pensar en términos de importancia para el usuario y no sólo en términos de qué funciones sería bueno tener.

2.1 Elección del lenguaje de programación y del entorno de desarrollo.

Para el desarrollo de este proyecto se compararon numerosas alternativas de desarrollo. Desde usar un único lenguaje de alta eficiencia como C++ y basarse en librerías intermedias para lograr portabilidad, hasta lenguajes que no dependieran en absoluto de la máquina de ejecución como puede ser Java. Pero sin perder de vista que para el objetivo fundamental de las operaciones también es necesario altas capacidades de cálculo en punto flotante y una eficiente gestión de la memoria (Lucas and Cabello, 2005).

Tabla 1. Comparación de lenguajes de programación

Criterio de selección	Peso	C++	C#	Java
Portabilidad	5	3	4	5
Capacidad 2D/3D	5	3	2	4
Matemática de precisión	5	3	2	2
Gestión memoria	5	5	3	3
Velocidad	10	10	8	6
Licencia	10	10	0	10
Eficiencia	10	10	6	6
Modularidad	5	4	4	4
Total		48	29	40

Los lenguajes que se evaluaron fueron Java, C++ y C#.

Para tomar una decisión en la elección del lenguaje, se realizó una tabla resumen (Tabla I) donde se le asignaron valores a los diferentes criterios de selección en función de su importancia para el desarrollo de la aplicación.

Teniendo en cuenta estos resultados se optó por escoger como lenguaje de programación el C++ para aprovechar su velocidad de ejecución, eficiencia y todas las potencialidades que ofrece de manera general.

2.2 Elección del entorno de programación.

El NVP es esencialmente una aplicación que no tiene una interfaz gráfica de usuario muy sofisticada, ya que trabajará como un proceso, recibiendo pedidos de usuarios, y controlando los procesos en ejecución de acuerdo a los requisitos descritos.

No obstante, y sobre todo para el cliente administrador, hay un grupo de funcionalidades que necesitan de una interfaz gráfica (por ejemplo, crear un modelo, listado de usuarios y procesos activos, etc.) que puede implementarse a través de un servicio web para acceso remoto, o localmente en el nodo.

En una primera versión se pensó en el acceso local, aunque en cualquier caso se necesita seleccionar un entorno que facilite la creación de dichas interfaces. Adicionalmente, se buscó un entorno de desarrollo integrado (IDE) que fuera una aplicación sobre software libre por las ventajas que conlleva. Entre los IDE que existen para Linux los más populares son KDevelop, NetBeans, Eclipse y Qtcreator.

La selección en este caso recayó sobre QtCreator pues tiene licencias tanto libre como comercial, desarrollada en C++ con añadidos. Es un paquete de herramientas maduro que cuenta aproximadamente con 500 clases, más de 9000 funciones y 500.000 líneas de código y que le da al programador mucha de la potencia que le brindan lenguajes como C# o Java con la eficiencia de código compilado en C++. Al igual que otros paquetes no sólo cuenta con clases para la construcción de interfaces de usuario, también incluye soporte para dibujo en 2D, hilos, red, bases de datos, etc.

2.3 Elección del sistema gestor de base de datos.

El NVP necesita almacenar información de forma permanente, relacionada con los modelos, los registros de actividades de cada usuario, la configuración general del sistema, etc. por lo que se requiere además de un gestor de bases de datos.

En la fecha de hoy las opciones principales son MySQL y PostgreSQL; PostgreSQL, además de ciertas mejoras, como el aumento del tamaño máximo de una fila, está poniendo especial énfasis en mejorar el rendimiento, habiendo conseguido en su versión 7.1.x grandes mejoras en este terreno. PostgreSQL además, en el caso de transacciones complejas, es mucho más estable y seguro que MySQL y la velocidad es comparable con la de MySQL. El nodo virtual de procesos deberá soportar una alta concurrencia en las consultas a la base de datos lo que exige estabilidad y velocidad en el gestor de base de datos. Por lo que el gestor de base de datos que más se adecúa es el PostgreSQL.

RESULTADOS

Propuesta de diseño de los elementos críticos del sistema.

1. Modelo del proceso y su solución.

El modelado es la fase, dentro del diseño de un sistema de control donde se busca el modelo matemático, que debe ser lo

más simple posible sin comprometer exactitud, dado que lo que se pretende es precisamente el acercamiento matemático más sencillo pero que arroje los resultados más fidedignos.

1.1 Soluciones numéricas.

Para la simulación de un sistema por computadora una vez obtenido el modelo, es necesario resolver las ecuaciones que lo componen. Para esto existe una variedad de métodos numéricos. Entre los más conocidos y los de mejores prestaciones se encuentran los llamados métodos de Runge – Kutta. Si se programan los algoritmos de Runge – Kutta y con ellos se resuelve el sistema de ecuaciones diferenciales que caracteriza la dinámica del proceso bajo simulación, solo restaría representar los datos de la manera adecuada en otro módulo o funcionalidad dentro del NVP. En el Nodo Virtual de Procesos se programaron los algoritmos Runge – Kutta de orden 2 y de orden 4.

1.2 Representación de las ecuaciones en el programa.

Contando con los métodos programados y la teoría general para la simulación se deben tener en cuenta que el nodo puede ir enriqueciendo su biblioteca de procesos de forma incremental, y que es deseable que esto se haga sin necesidad de modificar el programa o recompilarlo, de tal manera que los usuarios o clientes del nodo puedan introducir los modelos de interés sin esperar a nuevas versiones del software. Para poder satisfacer estos requisitos se debe:

- Definir una interface o plantilla general que describa las ecuaciones diferenciales de una forma que pueda ser interpretada de forma automática por el módulo correspondiente en el NVP.
- Brindar la posibilidad a los usuarios de incluir nuevos modelos matemáticos de procesos, sin la necesidad de recompilar toda la aplicación.

Para solucionar estas dificultades se idea un método simple y seguro utilizando librerías dinámicas. Las librerías dinámicas son recursos que brindan los sistemas operativos para incluir nuevas funcionalidades a los programas en tiempo de ejecución y hacer las aplicaciones más ligeras. No son más que un archivo que contiene código y datos, cuando estos se cargan, son incluidos en el espacio de código o datos del programa en cuestión y así este puede disponer de ellos.

Inicialmente el programa no contiene el código dentro de la librería, sino una referencia a esta. Semejante método brinda la posibilidad de que los usuarios creen sus propias librerías con sus modelos y las compilen, basándose en un patrón que se debe satisfacer. Con este sistema no hay límite de cantidad ni de tamaño de los modelos y se satisfacen ambos requisitos.

1.3 Librerías dinámicas como soporte de los modelos del Nodo Virtual de Procesos.

Una vez establecido que las librerías serán el soporte para los modelos en el Nodo Virtual de Procesos, se confecciona un

patrón a seguir por los usuarios a la hora de crearlas para que exista compatibilidad.

1.3.1 Librerías compartidas en GNU / Linux.

De manera general las referencias o las direcciones de las librerías donde se encuentran los modelos no son conocidas por el NVP, son cargadas en tiempo de ejecución y una vez realizada esta acción el programa conoce donde cargar el código que será utilizado por los métodos numéricos.

Este código no es ejecutado en la misma posición donde se carga, lo que significa que cada función puede ser llamada en diferentes momentos, en cualquier orden y por diferentes aplicaciones. Esto no es problema para el programador dado que el sistema operativo realiza estas tareas. Además la clase “QLibrary” de QtCreator provee los métodos para el manejo de estas acciones.

Cuando se define una librería en ambientes UNIX se define en un archivo de extensión “.so” o “.a”, que son archivos comunes. Además la declaración es sencilla como se muestra a continuación:

```
extern "C" double* f1 (double, *double, *double)
```

```
{//código de la función }
```

El especificador “extern C” impide que el compilador de C++ cambie el nombre de funciones declaradas en la librería.

1.3.2 Dll en Windows.

En Windows ocurre algo similar, las librerías compartidas son conocidas como Dll. En el momento de ser declaradas deben ser además explícitamente exportadas con la directiva de compilador “__declspec(dllexport)”.

Una declaración más portable, funciona de igual manera para un compilador de GNU o de Windows y queda de la siguiente manera:

```
extern "C" MY_EXPORT double f1 (double, *double, *double)
```

```
{ //código de la función }
```

Con MY_EXPORT definida como:

```
#ifdef Q_WS_WIN
```

```
#define MY_EXPORT __declspec(dllexport)
```

```
#else
```

```
#define MY_EXPORT
```

```
#endif
```

Donde “Q_WS_WIN” en el ambiente de QtCreator queda definida si se está construyendo el programa para ambientes Windows de Microsoft.

De esta manera se va completando el patrón que deben seguir los usuarios a la hora de crear las librerías con los modelos.

2. Conexión cliente-servidor en el nodo virtual de procesos.

El control de la conexión del nodo con las aplicaciones clientes se realiza mediante una clase controladora (CC_Conexion), en la cual se encuentran definidos todos los elementos necesarios para gestionar la comunicación.

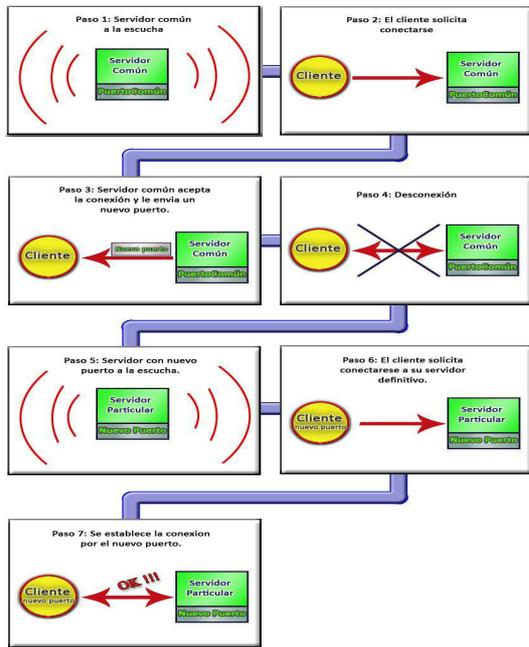


Figura 6. Diagrama en bloques del proceso de conexión.

El protocolo escogido fue TCP/IP, por ser de gran utilidad en el intercambio de información en redes locales e internet, por otra parte en las librerías de Qt existen clases como “QTcpSocket” y “QTcpServer” diseñadas para este tipo de comunicación.

Para el correcto funcionamiento del nodo y debido a la gran cantidad de valores que deben ser enviados a todos los usuarios durante la simulación, se decide, que la conexión se efectúa por un puerto diferente para cada cliente, evitando de esta manera congestión en la comunicación.

Como se puede observar en la Figura 6 el proceso de conexión de un nuevo cliente con el nodo se puede desglosar en siete pasos fundamentales:

- Paso 1: En el nodo se pone a la escucha un objeto servidor (QTcpServer), por el cual todos los clientes deben iniciar su

conexión (los clientes deben dominar la dirección IP y el puerto por el cual está escuchando este servidor).

- Paso 2: El cliente solicita la conexión al nodo a través de este objeto servidor (mediante un puerto común conocido).
- Paso 3: El objeto servidor acepta la conexión del cliente y le envía el puerto que le corresponde (uno diferente para cada cliente).
- Paso 4: Una vez que el cliente obtiene el puerto por el cual este debe conectarse con el nodo, entonces se desconecta de este objeto servidor.
- Paso 5: Se crea un nuevo objeto servidor en el NVP, que comienza la escucha por el nuevo puerto, por lo que es específico para cada cliente.
- Paso 6: A continuación el cliente trata de conectarse por el puerto que le fue enviado en el paso número 3.
- Paso 7: El servidor acepta la conexión y de esta manera cada cliente queda conectado al nodo a través de un puerto diferente.

Como se ve anteriormente la conexión de cada cliente al NVP, se realiza asignándole a cada cliente un puerto diferente.

3. Separación de las principales funciones del nodo en entornos de ejecución independientes.

En la figura 7 se muestra un esquema del sistema de hilos utilizado para separar las principales funciones del nodo en entornos de ejecución independientes. Como se puede observar se creó un hilo principal que se encarga de actualizar a los usuarios cada vez que se tiene un dato nuevo como resultado de la simulación. Por otra parte, cada vez que un proceso es activado; para realizar su simulación se crea un nuevo hilo el cual se ejecuta sin interferir en la simulación de cualquier otro proceso.

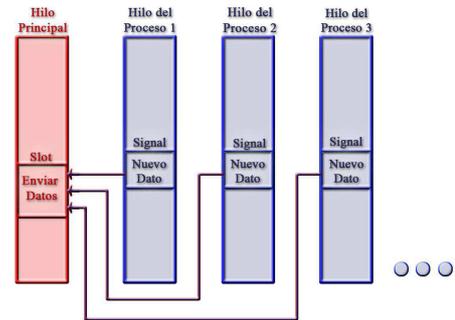


Figura 7. Esquema del sistema de hilos utilizado.

Como se muestra en la figura 7 para la comunicación entre los hilos encargados de la simulación de los procesos y el hilo principal (encargado de enviar los datos nuevos a cada uno de los usuarios) se efectúa mediante Signals y Slots.

Cada vez que en un hilo de proceso se obtenga un nuevo valor producto de la simulación se emite una señal que está conectada a un slot del hilo principal. De esta manera se logra independizar la simulación de la comunicación, evitando que se detenga la simulación para enviar los resultados de la misma. Para cumplir con el requisito de virtualización del nodo se aplicó el diseño mostrado mediante el cual se optimiza el uso de los recursos de hardware y aumenta el rendimiento y las prestaciones de la aplicación.

CONCLUSIONES

En este trabajo, teniendo en cuenta de la revisión de experiencias anteriores en el diseño y conceptualización de laboratorios y nodos virtuales de diverso tipo y las necesidades específicas de un nodo virtual que permita simular procesos de interés para el área de automatización, los autores han:

1. Determinado los requerimientos funcionales de un nodo virtual de procesos.
2. Seleccionado las plataformas de programación del sistema.
3. Descrito la forma de implementar las funcionalidades críticas (simulación del proceso, incorporación de modelos de procesos, conectividad y ejecución en entornos diferentes) del NVP.
4. Se ha desarrollado una primera versión de un nodo virtual de procesos donde se demuestra la viabilidad de las ideas propuestas.

REFERENCIAS

1. **AREA MOREIRA M.** “¿Qué aporta Internet al cambio pedagógico en la Educación Superior?”, en: Redes Multimedia y diseños virtuales. Actas del III Congreso Internacional de Comunicación Tecnología y Educación, Universidad de Oviedo, 2000.
2. **AREA MOREIRA M.**,” *Nuevas tecnologías, educación a distancia y la mercantilización de la formación*”. Revista Iberoamericana de Educación. ISSN: 1681-5653, 2004
3. **BETH CAMPBELL A., PARGAS R.**, “*Laptops in the Classroom. Technical Symposium on Computer Science Education*”, Proceedings of the 34th SIGCSE technical symposium on Computer science education Reno, Nevada, USA. 2004
4. **NEWBY G.**, “*Student Laptop Ownership Requirement and Centralization of Information Technology Services at a Large Public University. Annals of Cases on Information Technology*”, v5: 206-217. Hershey, Pennsylvania: Idea Group Publishing, 2001.
5. **CAPOTE MORALES M., GARCÍA ESPÓSITO A.**, “*Un estudio sobre las TICs y el alumnado universitario*”, ComByte 2004: Un nuevo sujeto para la sociedad de la información. Universidad de Las Palmas de Gran Canaria. Abril, 2004.
6. **GABINETE DE TELE-EDUCACIÓN.**” *Informe Asignaturas de Libre Elección mediante Teleformación curso 2003/2004*”, <http://www.gate.upm.es/evaluacion/documentos/informesevaluacion/Informe0304.pdf> . Universidad Politécnica de Madrid, 2004.

7. **DUART J., LUPIÁÑEZ F.**, “Estrategias en la introducción y uso de las TIC en la universidad”, Las TIC en la universidad: estrategia y transformación institucional [monográfico en línea]. Revista de Universidad y Sociedad del Conocimiento (RUSC). Vol. 2, núm. 1. Universidad Obrera de Cataluña (UOC). ISSN 1698-580X, 2005.
8. **ALBALADEJO BLÁZQUEZ A., GARCÍA QUINTANA A.**. Sesiones remotas vs acceso remoto. Laboratorio de prácticas libres ubicado en la EPS. Universidad de Alicante en España, 2005.
9. **MAIER S., HERRSCHER D.**,” *On Node Virtualization for Scalable Network Emulation*”, University of Stuttgart, Institute of Parallel and Distributed Systems (IPVS), Germany, 2005.
10. **VAQUERO J.**,”*Herramienta interactiva para la enseñanza y entrenamiento en la técnica de control predictivo*”, tesis en opción al grado de máster, Instituto Politécnico José Antonio Echeverría, La Habana, Cuba, 2007.
11. **NGUYEN, A.V, GILLET,D., SIRE, S.** “*Sustaining Collaboration within a Learning Community in Flexible Engineering Education*”, School of Engineering, Swiss Federal Institute of Technology (EPFL), Switzerland. 2004.
12. **DORMIDO, S.**, “*Proyecto AutomatL@bs. Red de laboratorios remotos en Automática*”, E.T.S. Ingeniería Informática, UNED, Madrid, 2007.
13. **VARGAS, H. ET AL.**, “*A systematic two-layer approach to develop web based experimentation environments for control engineering education*”, Dept. of Computer Science and Automatic Control UNED, Madrid, Spain, 2008.
14. **NGUYEN, A.V. ET AL.**. “*Sustaining the continuity of interaction in Web-based experimentation for Engineering Education*”. Proceedings of CALIE’04, Grenoble, France, 2004.
15. **RTAI-XML PROJECT TEAM**,”*RTAI-XML*”, <http://rtaixml.dsi.unifi.it/> , Appalachian State University, 2004.
16. **HOSSEINZAMAN A., BARGIELA A.**,”*ADA’s Virtual Node based Water System Simulator*” Department of Computing Nottingham Trent University Burton Street, Nottingham,1995.
17. **WU Y.**,” *A virtual node method for treatment of wells in modeling multiphase Flow in reservoirs*”, en: Twenty-Fourth Workshop on Geo-thermal Reservoir Engineering Stanford University, Stanford, California, 1999.
18. **MOLINO N., BAO Z.**,”*A Virtual Node Algorithm for Changing Mesh Topology During Simulation*”, Stanford University, 2004.
19. **MIYACHI T., CHINEN K.**, “*StarBED and SprinOS: Large scale general purpose network testbed and supporting software*”, Japan Advanced Institute of Science and Technology, Ishikawa, Japan, 2006.

AUTORES

Yalice Gámez Batista, Ingeniera en Automática, Profesora Asistente, Universidad de las Ciencias Informáticas km 5 ½ carretera San Antonio de los Baños, Rpto. Torrens, Boyero, Habana, 8372764, yaliceg@uci.cu. Sus intereses de investigación se relacionan con el desarrollo de plataformas virtuales para la simulación de procesos.

Valery Moreno Vega, Ingeniero en Máquinas Computadoras, máster en Informática Aplicada, Doctor en Ciencias Técnicas, profesor titular, Instituto Superior Politécnico José Antonio Echeverría, 266-3343 valery@electrica.cujae.edu.cu,

Actualmente realiza investigaciones en el área de robótica, informática aplicada a la automatización y en aplicaciones para el control utilizando métodos de inteligencia artificial.

Yoan Martínez Márquez, Licenciado en Idioma Inglés, Máster en ciencias, Profesor Asistente, Universidad de las Ciencias Informáticas km 5 ½ carretera San Antonio de los Baños, Rpto. Torrens, Boyero, Habana, 8372764, yoanm@uci.cu, Sus intereses de investigación se relacionan con el desarrollo de plataformas virtuales para la simulación de

procesos y los aspectos pedagógicos derivados de su aplicación práctica.