

Herramienta para la enseñanza en la carrera de Ingeniería Automática: Nodo Virtual de Procesos. Primera versión.

Ing. Yalice Gámez Batista¹, Dr. Valery Moreno Vega², MSc. Yoan Martínez Márquez³, Ing. Eliober Cleger Despaigne⁴, Ing. Daniel Hernández Garrigo⁵, Ing. Rolando Moreno Martínez⁶, Ing. Susana Gonce Fernández⁷, Ing. Mailen E. Escobar Pompa⁸

1 Universidad de las Ciencias informáticas, Cuba, yaliceg@uci.cu

2 Instituto Superior Politécnico José Antonio Echeverría, Cuba, valery@electrica.cujae.edu.cu

3 Universidad de las Ciencias informáticas, Cuba, yoanm@uci.cu

4 Universidad de las Ciencias informáticas, Cuba, ecleger@uci.cu

5 Universidad de las Ciencias informáticas, Cuba, dgarrigo@uci.cu

6 Joven Club de Computación, Cuba, rolando1402ad@pri.jovenclub.cu

7 Universidad de las Ciencias informáticas, Cuba, sgonce@uci.cu

8 Universidad de las Ciencias informáticas, Cuba, meescobar@uci.cu

Resumen

Tradicionalmente en la carrera de ingeniería Automática, 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. Este proyecto consiste en 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: Nodos Virtuales, Simulación, Control Automático, Procesos Industriales, Redes Virtuales

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 (Area, 2000).

El objetivo, a la hora de aplicar las TIC en la docencia, consta de (Area, 2000), (Area, 2004):

- 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.

Como señalan diversos autores (Beth and Pargas, 2004)(Newby, 2001), ya es un componente de excelencia universitaria el disponer de una infraestructura tecnológica que permiten 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 (Capote and García, 2004) se justifica sobradamente el apoyo que la mayor parte de las universidades está dando a la docencia virtual por Internet (Gabinete de Tele-Educación, 2004) en sus distintos modelos.

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 (Duart and Lupiáñez, 2005).

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 (Albaladejo and García, 2005). 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.

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 chequear los resultados teóricos tan pronto como la implementación sea posible. Estas mediciones usualmente se realizan para diferentes procesos o para procesos con diferentes condiciones.

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 correr en cualquier computadora. Esto se ajusta a las limitaciones económicas del centro.
- 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 (Maier and Herrscher, 2005). 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.

Antecedentes.

Las herramientas interactivas constituyen un elemento muy potente que permite reflejar la componente visual subyacente al control automático bajo la abstracción de conceptos

matemáticos, 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 (Vaquero, 2007).

En la actualidad existen muchos laboratorios virtuales y remotos que permiten realizar simulaciones de procesos industriales. Dentro de ellos se pueden mencionar Cockpit (Nguyen and Gillet, 2004), la red de laboratorios Automatl@b (Dormido, 2007), eElab (Vargas et al, 2008) montadas sobre el entorno de colaboración eMersion (Nguyen et al, 2004) y el RTAI-lab (RTAI-XML project team, 2004), 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.

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.

Para dar solución a esta problemática se crea un Nodo Virtual de Procesos que permite

ejecutar diferentes instancias del software en un único nodo (nodo físico) y cada instancia del software trabaja 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. A continuación se revisan algunas experiencias de nodos virtuales existentes y reportados por otros autores.

Posteriormente se describen los requisitos y funcionalidades del nodo virtual de procesos que se propone en este artículo, así como aspectos asociados a la 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 que ilustra el funcionamiento de acuerdo a los diferentes actores.

Softwares similares.

En la actualidad existen muchas aplicaciones que utilizan los nodos virtuales por las grandes ventajas que brindan en el aprovechamiento máximo de los recursos de las computadoras.

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” (Maier and Herrscher, 2005). Esta aplicación va dirigida a simular un entorno de redes configurable que permita reproducir un escenario real de cientos de nodos en comunicación. 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 (Hosseinzaman and Bargiela, 1995). Surge por la necesidad que tenían en la industria del agua de adquirir y almacenar datos de estaciones remotas para la inspección ingenieril. Este software tiene como tareas analizar los parámetros del sistema y llegar a decisiones que pueden ser aceptadas o

modificadas por el ingeniero responsable de estas operaciones. 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 (Wu, 1999). El método convencional de tratamiento de pozos geotermales o de reservas de petróleo no es lo suficientemente riguroso y puede dar lugar a soluciones físicamente incorrectas para las diferentes capas del pozo. 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. 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 (Molino and Bao, 2004). El algoritmo de nodo virtual determina automáticamente el número de réplicas así como la asignación de material para cada una. 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 (Miyachi and Chinen, 2006). Es una aplicación de pruebas basada en lotes de nodos que tiene como objetivos construir entornos de experimentaciones reales, complejos y de gran escala. Además se implementó “SpringOS” para soportar las simulaciones de acuerdo con la configuración de los usuarios. Ya ha sido probada con efectividad en diferentes experimentos.

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.

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.

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 modificarlo) o administradores.

La formalización de estas necesidades se establece mediante la descripción de requisitos. A continuación se presentan algunos de los requisitos que deberá cumplir el sistema NVP y que se constituyen en sus funcionalidades y características propias:

- Esta aplicación permitirá interactuar con los modelos de diferentes procesos, ya sea a través de simulaciones como probando aplicaciones en tiempo real
- 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.
- 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. 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. Estos clientes pueden trabajar como espectadores del proceso o podrán interactuar con una instancia del proceso al que se hayan suscrito.

- 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.

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). 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.

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

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.

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 estas se destacan:

- Evita la dependencia tecnológica de empresas foráneas.
- Ahorros por pagos de licencias de softwares.
- Posibilidad de revisar el código fuente.

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.

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.

Modelo matemático de los procesos.

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.

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. Estos métodos son conocidos como incrementales, pues calculan el resultado siguiente basándose en el anterior utilizando un incremento fijo en la abscisa.

La solución que arrojan estos métodos, no es más que el valor que tomaría la función solución $y = y(x)$ para un punto del eje x , partiendo siempre de una condición inicial para calcular el primer resultado. Si se programan los algoritmos de Runge – Kutta y con ellos se resuelve un sistema de ecuaciones diferenciales, para incurrir en la simulación computarizada solo quedaría representar los datos de la manera adecuada. En el Nodo Virtual de Procesos se programan los algoritmos:

- Runge – Kutta de orden 2;
- Runge – Kutta de orden 4;

Representación de las ecuaciones en el programa.

Contando con los métodos programados y la teoría general para la simulación se solicita cumplir con los siguientes requisitos:

Disponer de la forma de escribir las ecuaciones diferenciales de manera entendible para el programa y el algoritmo, puesto que los algoritmos necesitan invocar las ecuaciones del modelo para encontrar la solución.

Satisfacer una de las funcionalidades del programa, que es 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.

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

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.

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.

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. El protocolo escogido fue TCP/IP, por ser de gran utilidad en el intercambio de información en redes locales e internet22, 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.

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.

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

La filosofía del nodo virtual de procesos se fundamenta principalmente en la simulación paralela de varios procesos concurrentes sin que intervengan unos con otros. Una manera de regular el acceso a recursos de hardware

exclusivos de un determinado número de consumidores es la virtualización de nodos. En este caso los consumidores son los entornos de ejecución independientes para cada proceso, que se encuentran sujetos a las propiedades de la simulación.

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.

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.

Como se puede constatar, la conectividad del sistema está asegurada por la estructura de clases mostrada anteriormente que brinda el soporte necesario para lograr la completa funcionalidad del nodo en la red. 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

Con este trabajo se demostró la necesidad de la implementación de una aplicación Nodo Virtual de Procesos que permita, a los estudiantes de la carrera de Automática en el Instituto Superior Politécnico José Antonio Echeverría, poner en práctica los conocimientos teóricos que reciben en la misma. Se determinaron los requerimientos

funcionales de la aplicación así como los elementos necesarios para su implementación.

Se obtuvo como resultado una primera versión de la aplicación, donde se implementaron los métodos numéricos Runge – Kutta de orden 2 y - Runge – Kutta de orden 4 y se usó como mecanismo para la incorporación de los modelos matemáticos librerías dinámicas. Se utilizaron sockets para la comunicación entre el servidor y los clientes y se definió una separación entre las principales funciones para el trabajo con lo hilo de ejecución. De esta manera se dio solución parcial a la problemática planteada.

Referencias:

- Albaladejo Blázquez A., García Quintana A. (2005). Sesiones remotas vs acceso remoto. Laboratorio de prácticas libres ubicuo en la EPS. Universidad de Alicante en España.
- Area Moreira M. (2000). “¿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, septiembre, pp. 128-135.
- Area Moreira M. (2004),” Nuevas tecnologías, educación a distancia y la mercantilización de la formación”. Revista Iberoamericana de Educación. ISSN: 1681-5653.
- Beth Campbell A., Pargas R.(2004), “Laptops in the Classroom. Technical Symposium on Computer Science Education”, Proceedings of the 34th SIGCSE technical symposium on Computer science education Reno, Nevada, USA.
- Boullón Garzón R., Rodríguez Laredo P.(2006), “Comparativas SQL”, <http://www.intitec.com/varios/ComparativaSQL.pdf>
- Capote Morales M., García Espósito A. (2004), “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.

- Dormido, S. (2007), "Proyecto AutomatL@bs. Red de laboratorios remotos en Automática", E.T.S. Ingeniería Informática, UNED, Madrid.
- Duart J., Lupiáñez F. (2005), "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.
- Gabinete de Tele-Educación (2004). "Informe Asignaturas de Libre Elección mediante Teleformación curso 2003/2004", <http://www.gate.upm.es/evaluacion/documentos/informesevaluacion/Informe0304.pdf>. Universidad Politécnica de Madrid.
- Gámez Batista, Y. and Moreno Vega, V. (2008), "Consideraciones generales para el diseño de una herramienta interactiva de simulación de procesos", en Conferencia Internacional FIE'08, Santiago de Cuba, 2008.
- Furmankiewicz, J. (2008). Eclipse, NetBeans, and IntelliJ: Assessing the Survivors of the Java IDE Wars, <http://www.deux.com/Java/Article/34009/0>
- Hosseinzaman A., Bargiela A. (1995), "ADA's Virtual Node based Water System Simulator" Department of Computing Nottingham Trent University Burton Street, Nottingham
- Lucas Rodríguez F., Cabello Quintero A. (2005), "Simulación de procesos de información y criptografía cuántica" Departamento Física Aplicada I, Sevilla.
- Maier S., Herrscher D. (2005), "On Node Virtualization for Scalable Network Emulation", University of Stuttgart, Institute of Parallel and Distributed Systems (IPVS), Germany.
- Miyachi T., Chinen K. (2006), "StarBED and SprinOS: Large scale general purpose network testbed and supporting software", Japan Advanced Institute of Science and Technology, Ishikawa, Japan.
- Molino N., Bao Z. (2004), "A Virtual Node Algorithm for Changing Mesh Topology During Simulation", Stanford University.
- Newby G. (2001), "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.
- Nguyen, A.V. et al. (2004). "Sustaining the continuity of interaction in Web-based experimentation for Engineering Education". Proceedings of CALIE'04, Grenoble, France.
- Nguyen, A.V, Gillet, D., Sire, S. (2004) "Sustaining Collaboration within a Learning Community in Flexible Engineering Education", School of Engineering, Swiss Federal Institute of Technology (EPFL), Switzerland.
- Placeres Hernández J., De León Rosales L. (2006), "Algunos rasgos de la Universidad de hoy", Revista Avanzada científica, vol 9 no 3/2006, issn 1029-3450
- RTAI-XML Project team (2004), "RTAI-XML", <http://rtaxml.dsi.unifi.it/>, Appalachian State University,
- Vaquero J. (2007), "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.
- Vargas, H. et al (2008), "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
- Wu Y. (1999), "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, January 25-27