

Sistema Distribuido en Tiempo Real

Julio Javier Castillo

jotacastillo@gmail.com

Marina Elizabeth Cardenas

angelaesmeralda@gmail.com

LIS-Laboratorio de Investigación de Software
Dpto. Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Maestro M. López esq. Cruz Roja Argentina, Córdoba - Argentina

Abstract

Nuestra línea de investigación consiste en emplear diferentes metodologías de modelado de software para el modelado de sistemas en tiempo real y distribuidos. Trabajamos para ello sobre un sistema que hemos construido a tal efecto y sobre el cual aplicaremos diferentes metodologías de modelado continuando lo realizado en [1] y [2].

El sistema en concreto es un control distribuido en tiempo real de un horno eléctrico. Dicho control se lleva a cabo en forma remota sobre el protocolo tcp/ip a través del empleo de un webservice. Para la interfaz electrónica de control se emplea microcontroladores y se utiliza Microsoft VisualC#.Net 2005 para el control desde una computadora PC a lazo cerrado con el sistema de control de temperatura.

Se estudian también las técnicas de control distribuidas necesarias y más eficientes para el comando remoto de un sistema de control genérico.

Este trabajo se modela mediante la metodología de desarrollo de software RUP (Rational Unified Process [Booch, Jacobson 1999]) para la construcción de un sistema distribuido de tiempo real semi-sof, time driven (distributed semi-soft real time system) y actualmente se está trabajando para modelarla con la metodología COMET.

Como resultados de este trabajo podemos observar cómo emplear RUP para el diseño de sistemas en tiempo real y como interactúan embedding software – computer software – web software en una aplicación concreta.

Palabras claves: RUP, COMET, distributed semi-soft real time systems, furnace control, PID-Control.

Introducción:

En concordancia con el trabajo presentado en el Workshop de Investigadores de Ciencias de la Computación 2007, en Trelew-Chubut, denominado “Un Sistema de Tiempo Real Distribuido Semi-Soft usando RUP”[1] hemos continuado con nuestra línea de investigación aplicando técnicas de modelado de Ingeniería del Software que nos permitan modelar en forma certera nuestro sistema.

Inicialmente se utilizó la metodología RUP(Rational Unified Process [Booch, Jacobson 1999]) para modelar este sistema.

Sin embargo, debido a su generalidad y poca especialización para modelar sistemas en tiempo real, hemos optado por modelar este sistema paralelamente con la metodología COMET (Concurrent Object Modeling and Architectural Design Method) el cual también emplea UML como lenguaje de modelado.

El grupo de investigación cree que esta metodología es mucho más específica que RUP, para modelar sistemas con alto grado de concurrencia, paralelismo, en ambientes distribuidos y sistemas en tiempo real en general.

Es tema de nuestras futuras investigaciones el determinar fehacientemente cual de estas dos metodologías es más adecuada para el modelado de sistemas con estas características, determinando en que casos es conveniente la adopción de cada una de ellas.

Por medio de este trabajo se pretende realizar un control de un horno industrial eléctrico en forma remota a través del diseño de un “Semi-Soft Real Time System – Time Driven”, empleando para ello el RUP (Rational Unified Process) como proceso de desarrollo de software y utilizando UML (Unified Modeling Language) como lenguaje de modelado.

Decimos que este sistema es “Time-Driven” ya que las acciones del sistema son dirigidas principalmente por el pasaje de un intervalo de tiempo a otro. Este sistema ejecuta en forma periódica ciertas tareas, en base a la ocurrencia de “deadlines” en el tiempo.

RUP se ha empleado exitosamente para el modelado de sistemas de los más diversos dominios, sin embargo RUP presenta algunas limitaciones cuando se modela sistemas en tiempo real. Por ejemplo, en la fase de diseño se debe decir entre el “diseño de componentes en tiempo real” (generalmente empleando Rational Rose-RT) y “el diseño de componentes que no son de tiempo real” (generalmente empleando Rational Rose como herramienta de modelado), pero RUP no provee de guías de cómo realizar esta elección, ni expresa los beneficios y las consecuencias de estos dos métodos de diseño.

Debido a nuestro dominio de problemas, frecuentemente preferiremos el uso del inglés para las palabras técnicas.

El presente trabajo se divide en varias secciones: Metodología, Estado Actual del Proyecto, Cadena de Adquisición, Conclusiones y Referencias.

Metodología

Se ha decidido emplear RUP como metodología de diseño de desarrollo de software debido a su amplia aceptación en el ámbito industrial y académico, ya que brinda un marco de trabajo genérico que se puede adaptar para modelar sistemas en tiempo real distribuidos como es el caso de este proyecto.

También se ha decidido emplear paralelamente la metodología COMET para el modelado de este sistema, ya que esta metodología fue concebida para modelar sistemas con alto grado de concurrencia y fue pensada principalmente para ambientes distribuidos.

Se ha comenzado a aplicar COMET paralela e independientemente, sobre el sistema que ya ha sido modelado en gran parte con RUP. Se pretende entonces comparar ambas metodologías de diseño de desarrollo de software y determinar cual de las dos es más conveniente para el modelado de este sistema en particular, y trataremos de extrapolar las conclusiones a sistemas distribuidos en tiempo real en general, proponiendo guías que permitan optar por una u otra metodología.

A continuación resumimos algunos de los alcances más importantes del sistema:

- Brindar información acerca de la temperatura actual del horno en una interfaz gráfica de usuario.
- Registrar la temperatura del horno durante un determinado intervalo de tiempo.
- Mostrar en forma gráfica un historial de variación de temperatura de los últimos k-cambios de temperatura registrados.
- Permitir al usuario local/remoto encender el horno hasta llegar a una temperatura que él desee.
- Permitir al usuario local/remoto apagar el horno cuando él lo desee.

- Permitir al usuario local/remoto realizar una planificación del comportamiento del horno eléctrico.
- Establecer los límites mínimos y máximos dentro de los cuales se considerará seguro el sistema de control de temperatura.
- Permitir la elección de un conjunto de usuarios(lista de usuarios) a los cuales se les informará por medio de un envío de e-mail, los cambios de temperatura que sean considerados críticos o alarmantes.
- Brindar al usuario la opción de elegir diferentes mecanismos de control de temperatura.

Como requisitos no funcionales podemos destacar:

- Informar a los usuarios de la “Lista de Alertas” acerca de las variaciones de temperatura fuera de los límites de control en un tiempo inferior a un minuto.

En [1] y [2] se pueden consultar el diagrama de contexto y el diagrama de caso de uso del sistema de información, del “sistema distribuido en tiempo real semi-soft” que realiza un control remoto de la temperatura.

Estado Actual del Proyecto

Actualmente el sistema ha completado completamente la fase de inicio y la fase de elaboración, ubicándose en la tercera iteración de la fase de construcción de la metodología RUP. Es por ello que ya se ha conseguido una línea base de la arquitectura y ya se han mitigado los riesgos más importantes que podrían hacer peligrar este trabajo o proyecto.

Se encuentra funcionando el control de temperatura mediante el algoritmo de control PID (Proporcional Integral Derivativo) basado en PWM(Pulse Width Modulation) [9]. También está en funcionamiento el control remoto desde Internet, y es posible establecer en forma remota(desde la web) la temperatura óptima, temperatura máxima, y temperatura mínima, del horno de temperatura y adjuntar una lista de direcciones de correo electrónico al cual le llegarán mensajes de texto cuando el sistema se encuentre fuera de los límites permitidos, es decir por debajo de la temperatura mínima, o por encima de la temperatura máxima. La temperatura deseada será el “set-point” del algoritmo PID.

Se realizaron iteraciones a través de los 5 flujos de trabajos genéricos (workflows).En la fase de Inicio se realizaron dos iteraciones cuyo objetivo fue la investigación del “state of the art” de los “distributed soft/hard real time systems”, “intelligence controllers”, “distributed controllers”, entre otros. En esta etapa se identificaron y priorizaron los riesgos más importantes y se desarrolló un modelo preliminar de Uses Cases con los “Critical Uses Cases”, obteniendo de esta manera un esbozo de la arquitectura del sistema.

Por otra parte, la fase de elaboración constó de tres iteraciones en las que se especificaron con mayor nivel de detalle los use cases obtenidos en la fase de inicio y se diseñó la arquitectura del sistema. A su vez en la primera iteración de esta etapa se comenzó a implementar y a probar el “embedded software” y en la tercera iteración se comenzó a desarrollar la electrónica de control del horno.

A continuación se presenta un modelo de “Acquisition Chain”, que ilustra como se lleva a cabo el control en forma remota de este “semi-soft real time system”.

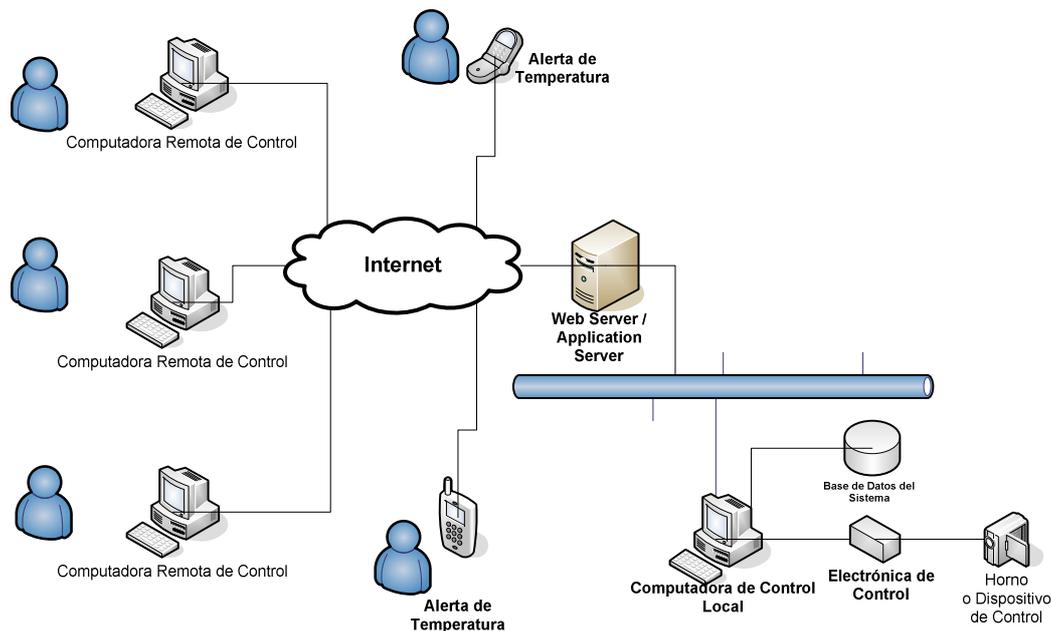


Figura 1. Cadena de Adquisición de alto nivel.

Cadena de Adquisición :

A continuación se describen sus principales elementos (Véase figura 1):

- Horno eléctrico de temperatura.
- Conversor ADC de 10 bits , parte del PIC 16F877A
- Interfaz USART – RS232.
- PC
- Web-Server, con Webservices
- Web-Client / Remote Control

La temperatura es sensada a través de un sensor de temperatura LM35, el cual entrega un valor de tensión que es proporcional a la temperatura medida en el horno eléctrico.

Este valor es tomado por el PIC y pasado a la PC a través del puerto serie COM1 empleando el protocolo RS232.

Si la temperatura se encuentra fuera de los límites mínimos/máximos establecidos, entonces se envía un e-mail de alerta a una lista de usuarios.

Además, es posible visualizar gráficamente el historial de variación de temperatura de las últimas k-medidas efectuadas.

Por otra parte, la PC permite realizar dos tipos de controles :

- control del tipo SI/NO y - control PID(Proporcional Integral Derivativo).

Conclusiones

Se está desarrollando una aplicación exitosa y funcional de RUP para el modelado de un sistema distribuido semi-soft de tiempo real, en el cual se han empleado conocimientos de distintas áreas, tales como ingeniería del software, electrónica, y ciencias de la computación.

Sin embargo, este sistema esta sujeto a muchas mejoras y ampliaciones como por ejemplo la aplicación de técnicas de tunning pid, fuzzy logic, entre otras. Estas técnicas se estudiarán y se aplicarán en el presente trabajo en un futuro cercano.

El Sistema de Control Automático de Temperatura del Horno Eléctrico diseñado presenta características novedosas que lo convierten en un sistema de tiempo de real blando, cuyas variación de temperatura fuera de los límites parametrizados y previamente establecidos, permite a un usuario ubicado en cualquier parte del mundo enterarse de tal evento con una demora en el caso promedio de unos 50 segundos.

Por otra parte este sistema es susceptible de grandes mejoras de funcionalidad y performance, entre tales mejoras citaremos la generalización a un control de k-dispositivos simultáneos, o la adición de un mecanismo de enfriamiento de la temperatura para realizar de una manera más versátil el control de la temperatura.

Además sería posible trabajar con hornos de mayor temperatura y aplicarles este mismo sistema de control con mínimos cambios.

Este trabajo tuvo como principal motivación la necesidad de integrar diferentes tecnologías en pos de construir un sistema de tiempo real de control de temperatura empleando diferentes metodologías de diseño de desarrollo de software, y que sea capaz de informar a un conjunto de personas responsables los momentos en que el sistema esté fuera de control o bien cuando se haya llevado el sistema fuera de los límites preestablecidos.

Referencias

- [1] CASTILLO Julio, CARDENAS Marina, *Un Sistema de Tiempo Real Distribuido Semi-Soft usando RUP*. WICC 2007 Trelew - Argentina.
- [2] CASTILLO Julio, CARDENAS Marina, *Sistema en Tiempo Real Distribuido Semi-Soft empleando RUP*. JIDIS 2007. Córdoba - Argentina.
- [3] JACOBSON. I; BOOCH, G. y RUMBAUGH J. *El Proceso de Desarrollo de Software*.
- [4] JACOBSON. I; BOOCH, G. y RUMBAUGH J. *UML, El Lenguaje Unificado de Modelado*.
- [5] BOOCH, G. (1996) *Object Oriented Design with Applications*. 2ª ed. Wilmington, Addison-Wesley Iberoamericana.
- [6] JACOBSON, I. y otros. (1994) *Object-Oriented Software Engineering. A Use Case Driven Approach*. Estados Unidos de América, Addison-Wesley/ACM press.
- [7] RUMBAUGH, J. y otros. (1991) *Object-Oriented Modeling and Design*. Englewood Cliffs, Prentice-Hall.
- [8] MARTIN FOWLER, KENDALL SCOTT. (1999) *UML Gota a Gota*, Addison Wesley.
- [9] COURRIOU JEAN-PIERRE, *Process Control: Theory and Applications* , Springer
- [10] RICH, E. y KNIGHT, K. (1995) *Inteligencia Artificial*. McGraw Hill.
- [11] OMG – www.omg.org – artículos varios.
- [12] HASSAN GOMMA, “*Designing Concurrent, Distributed, and Real-Time Applications with UML*”. Addison-Wesley.