

Abstract

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

Introducción:

Diariamente nos presentamos con problemas de mantenimiento de temperatura, tal es el caso de mantener dentro de un determinado intervalo la temperatura del vehículo que estamos conduciendo, o bien de mantener dentro del rango permitido la temperatura del microprocesador de nuestra PC.

El cuerpo humano es otro ejemplo de un sistema de control y regulación de la temperatura, el cual debe mantenerse entre 36.4°C. y 37.1°C. dependiendo de la actividad que se realice, pero cuando esta desciende o aumenta considerablemente es señal de alarma para el organismo.

El cuerpo mantiene la temperatura constante mediante un centro conocido como termorregulador, localizada en la parte del cerebro llamada hipotálamo, que constituye en "kernel" del sistema de control del cuerpo humano.

Generalmente, la temperatura suele aumentar ante una infección o enfermedad (noisy environment) y suele descender cuando hay algún traumatismo (noisy environment). De esta manera, con una temperatura por debajo de los 35°C tenemos un cuadro de hipotermia, y por encima de los 38 tenemos fiebre.

Como podemos observar, estamos rodeados de situaciones en donde el control de la temperatura es una variable vital que debe y tiene que ser controlada. Además, el prescindir de este control puede causar graves pérdidas materiales, percances económicos y hasta pérdida de vidas humanas, u otro tipo de daños.

Por otra parte, a nivel industrial, el control de la temperatura es una variable de gran relevancia, requiriendo generalmente de sistemas en tiempo real duros (hard real time systems) que realicen el control de estos procesos.

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.

Mediante este trabajo se podrá evidenciar como el RUP puede ayudar al diseño de sistemas en tiempo real. Para poder llevar a cabo esta labor, fue necesario realizar algunas adaptaciones a RUP en el modelo de diseño para que modele de

manera efectiva este tipo de sistemas software.

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.

La idea inicial de este trabajo era la de realizar el control de un horno industrial a gas, sin embargo, debido a la imposibilidad de realizar pruebas sobre un sistema de éstas características, se tomó como prueba un horno eléctrico que se utiliza para la esterilización de materiales.

Como aclaración cabe destacar que se empleará el idioma inglés para los términos técnicos, dejando el uso del español sólo para los casos en que se lo ha considerado conveniente, debido a que en estos dominios, los nombres técnicos en inglés son mejor conocidos y aceptados por la comunidad científica internacional.

El presente trabajo se divide en varias secciones : Metodología, Estado Actual del Proyecto, Bloques de Construcción, Algoritmo de Control, Desarrollo del Software de Control, Cadena de Adquisición, Conclusiones, Agradecimientos 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.

Debido a la extensión máxima del paper, no será posible transcribir todos los artefactos que se obtienen al aplicar la metodología RUP. Por ello, se han elegido los Criticals Uses Cases para representar las funcionalidades más importantes del ciclo de vida del desarrollo del sistema software.

Además se provee de una descripción en bloques funcionales: Bloque Físico, Bloque de Electrónica de Control y Bloque de Sistema Software de Control.

A continuación resumimos algunos de los alcances más importantes:

- 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.

Este sistema ha sido desarrollado teniendo en cuenta los siguientes principios de Ingeniería del Software: Adaptabilidad, Usabilidad, Reusabilidad, y Separación de Incumbencias.

Estado Actual del Proyecto

Actualmente el sistema ha completado completamente la fase de inicio y la fase de elaboración, ubicándose en la segunda iteración de la fase de construcción. 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 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 remote control de este semi-soft real time system.

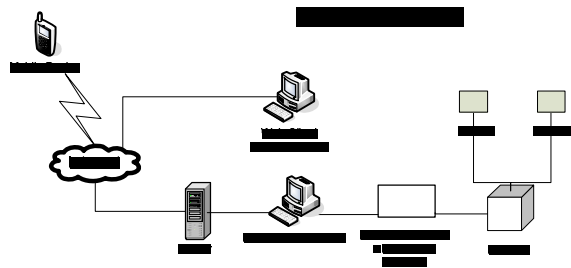


Figura 1. Acquisition Chain de alto nivel.

Requeriments WorkFlow – Flujo de Trabajo de Requisitos :

En este workflow se elaboró un context model (modelo de contexto) lo cual provee al sistema de un nivel de abstracción mayor que el Use Case Model a nivel de sistema de información.

A continuación se adjunta el diagrama:

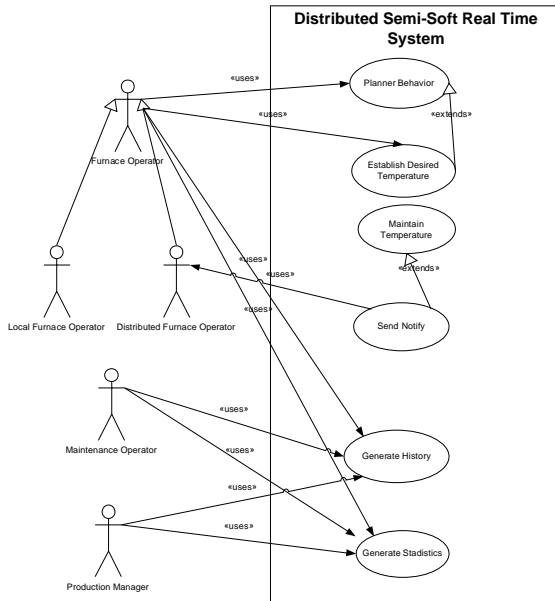


Figura 2. Context Model del Soft-Real Time System – simplificado- mostrando los critical uses cases.

Breve descripción de los Use Cases:

Planning Behavior, permite al operador del horno establecer una planificación del funcionamiento del horno determinando máximos y mínimos permitidos, lapsos de tiempos de las temperaturas deseadas, también se puede establecer la temperatura deseada.

Establish Desired Temperatura, permite al operador del horno establecer la temperatura deseada y el intervalo de tiempo válido para el mismo, como así también una lista de distribución indicando las personas a las que se les notificará cuando acontezca una variación de la temperatura fuera del límite previsto.

Send Notify, envía una notificación mediante un sms a un dispositivo móvil. Esta notificación informa acerca de una variación de la temperatura fuera de los rangos permitidos para la misma.

Show History, muestra un historial acerca de la variación de la temperatura a través de una carta de control, para poder

determinar las causas comunes y especiales que influyen en el proceso de control de temperatura y que pueden llevar a que el proceso no sea estable.

Generate Statistics, permite la generación de estadísticas mostrando información sumariada acerca del historial de variación de temperatura.

En este workflow también se elaboró un Use Case Analysis Model mediante el cual se pueden identificar los actores de este Distributed Semi-Soft Real Time System y los Use Case que proveen su funcionalidad.

A continuación se adjunta el diagrama:

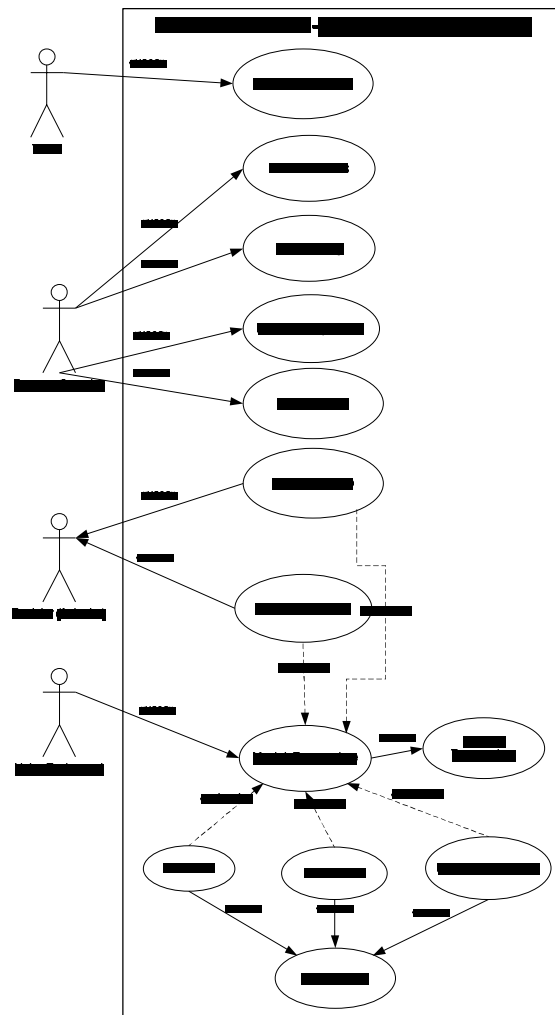


Figura 3. Use Case Model del Information System del Soft-Real Time System – simplificado - mostrando los critical uses cases.

Los nombres de estos uses cases son muy descriptivos, razón por la cual no se provee de un resumen.

Bloques de Construcción:

A continuación se enumeran y describen los bloques funcionales de este sistema en tiempo real distribuido.

Ø Bloque Físico:

Este bloque está constituido por un Horno Eléctrico empleado para la esterilización de materiales quirúrgicos.

El Horno puede soportar temperaturas que oscilan entre los 0°C y 140°C, aunque en este sistema solo trabajaremos dentro del intervalo de 15°C a 94°C.

Ø Bloque de Electrónica de Control :

Este bloque esta conformado por una:

- Tarjeta de adquisición de datos, que permite la comunicación con la PC y es el medio a través del cual es posible obtener la temperatura del horno, como así también ejercer una acción de control sobre el mismo para poder elevar su temperatura.

Nótese que en la presente implementación no es posible enfriar automáticamente el horno eléctrico con medios electro-mecánicos.

El componente principal que permite la comunicación PC – Mundo Exterior es un Microcontrolador PIC 16F877A, el cual es debidamente programado para que cumpla las tareas de adquirir los datos (sensar la temperatura), como así también de activar un actuador para que modifique la temperatura del horno eléctrico.

Ø Bloque de Software de Control del Distributed Semi-Soft Real Time System:

Este bloque de control presenta 3 elementos constitutivos, cada uno con un nivel de abstracción diferente y que corresponden a 2 diferentes paradigmas de programación:

- Programa Pic 16F877A : que corresponde al Paradigma Estructurado (Lineal).
- Programa Aplicación PC: programado en C#, que corresponde al Paradigma Orientado a Objetos.
- Programa WebAplicattion sobre un Web-Server (PC): web-software que corresponde a un Paradigma Orientado a Objetos y que responde al patrón de arquitectura Cliente-Servidor de 3 capas.

• Acquisition Chain / Cadena de Adquisición :

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

- 1) Elemento a controlar: Horno eléctrico de temperatura.
- 2) Conversor ADC de 10 bits (de los cuales solo se emplearán 8 bits)
- 3) Interfaz USART – RS232.
- 4) PC
- 5) Web-Server, con Webservices
- 6) 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 tomando por el PIC y pasado a la PC a través del puerto serie COM1 empleando el protocolo RS232.

El valor de temperatura es registrado por la PC para poder realizar con ellos análisis posteriores.

La PC determina si la temperatura se encuentra fuera de los límites mínimos/máximos establecidos y cuando esto sucede 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-mediciones efectuadas.

Por otra parte, la PC permite realizar un control del tipo SI/NO (que es un algoritmo de control trivial), sobre el horno de temperatura, como así también permite realizar un control empleando el algoritmo PID (Proporcional Integral Derivativo).

Algoritmo de Control:

Es un algoritmo de control que se emplea en el control de procesos industriales, posee tres acciones de control proporcional-integral-derivativo, de allí su nombre.

Este algoritmo constituye el kernel del sistema de control a lazo cerrado:

$$\text{Output}(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de}{dt} \right)$$

Donde:

- $e(t) = \text{Setpoint} - \text{Measurement}(t)$, es la señal del error.
- K_p , T_i , T_d : son constantes que son usadas para el afinamiento del lazo de control.
- K_p : Proporcional Gain: Se corrige en forma proporcional a la desviación.
- T_i : Tiempo Integral
- T_d : Tiempo Derivativo

La PC controla la acción de apagado/encendido del horno a través del Microcontrolador.

El Microcontrolador realiza un control SI/NO sobre la alimentación del horno, y de esta manera puede elevar su temperatura (es decir, se enciende el horno suministrándole energía eléctrica) o bien disminuirla (es decir, se apaga el horno y se debe esperar a que el mismo disipe el calor).

• Desarrollo del Software de Control:

El sistema software de control se desarrolla en assembler en el caso del PIC, en C# en el caso de la Windows Application, y en ASP.Net y C# en el caso de la Web Application y Webservice.

Se han integrado diferentes lenguajes de programación para poder alcanzar el objetivo inicialmente planteado.

Cabe notar que el microcontrolador está siendo empleado solamente como un dispositivo “proxy” entre la PC y el horno de temperatura, es decir empleamos el PIC como un dispositivo intermediario que nos facilite la comunicación y el control del actuador (resistencia de temperatura). En este sentido lo más importante del PIC será su conversor ADC de aproximaciones sucesivas de 10 bits, el cual es configurado para trabajar en modo 8 bits. Se prevee también una modificación en el control, de modo que el microcontrolador tenga mayor carga operacional implementando él mismo el control PID.

El software de la Windows Application será la encargada de mostrar la temperatura actual del horno, y su historial de variación. Se muestra en forma gráfica para lograr una comprensión más fácil por parte del usuario.

Finalmente, el software de la Web Application será el encargado de enviar el mensaje de alerta hacia una lista de distribución previamente definida, y de brindar una interfaz a través de un webservice, a un operador del horno que puede estar localizado a cualquier distancia del dispositivo controlado.

Conclusiones

Se ha mostrado 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.

En cuanto a su fiabilidad, este sistema a pequeña escala ha sido probado múltiples veces, entre ellas, se lo probó durante 48 horas seguidas, introduciendo variaciones controladas en el ambiente como ventilación o añadiendo más temperatura, y durante este tiempo no se ha evidenciado ningún error ni desfase mayor a 1,5°C grados centígrados respecto de la temperatura deseada, esta prueba se realizó a 67°C. Se realizaron otras pruebas a 50°C, 60°C, 70°C, 80°C y 90°C, y en todos los casos se obtuvieron resultados similares.

Este sistema está sujeto a muchas mejoras y ampliaciones como por ejemplo la aplicación de técnicas de *tuning* 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 variaciones 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 RUP 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.

Agradecimientos

A los doctores Francisco Tamarit, y Carlos Marquez, y al Ing. Walter Zaninetti que fueron de guía para este extenso trabajo. Parte de este trabajo es parte de la tesis de grado en Lic. en Computación del A.C. Julio Castillo en la FAMAF.

Referencias

- q Jacobson, I ; Booch, G. y Rumbaugh J. *El Proceso de Desarrollo de Software*.
- q Jacobson, I ; Booch, G. y Rumbaugh J. *.UML , El Lenguaje Unificado de Modelado*.
- q BOOCH, G. (1996) *Object Oriented Design with Applications*. 2ª ed. Wilmington, Addison-Wesley Iberoamericana.
- q JACOBSON, I. y otros. (1994) *Object-Oriented Software Engineering. A Use Case Driven Approach*. Estados Unidos de América, Addison-Wesley/ACM press.
- q RUMBAUGH, J. y otros. (1991) *Object-Oriented Modeling and Design*. Englewood Cliffs, Prentice-Hall.
- q MARTIN FOWLER, KENDALL SCOTT. (1999) *UML Gota a Gota*, Addison Wesley.
- q Jean-Pierre Courriou, *Process Control: Theory and Applications*, Springer
- q Rich, E. y Knight, K. (1995) *Inteligencia Artificial*. McGraw Hill.
- q OMG – www.omg.org – artículos varios.
- q *Papers varios*.

