

Enseñanza de Software de Sistemas de Tiempo Real.

Ing. A.De Giusti¹ , Lic. M. Naiouf² , Lic. L. Lanzarini³

*Laboratorio de Investigación y Desarrollo en Informática⁴
Departamento de Informática - Facultad de Ciencias Exactas
Universidad Nacional de La Plata*

Resumen

Se discute la experiencia de los autores dictando durante varios años un curso de Software de Sistemas de Tiempo Real, en el que han experimentado diversos enfoques teóricos y prácticos.

Se analizan las dificultades al estructurar inicialmente el curso (en 1991) que estaban centradas en la necesidad de definir un contenido teórico de interés para alumnos de Informática, y sobre todo en estructurar una práctica coherente a partir de los conocimientos previos y dentro de la escasa disponibilidad de recursos específicos.

Posteriormente se resume la evolución de los contenidos teóricos y las diferentes herramientas sobre la que los alumnos han efectuado experiencias prácticas, analizando su impacto pedagógico. Se trata de mostrar claramente el alcance de los diferentes enfoques (centrados en lenguajes, centrados en el control de tiempo real, centrados en la especificación), las dificultades encontradas y las soluciones adoptadas.

Por último, se analiza el contexto de las asignaturas previas y de aplicación posterior y se menciona una serie de "problemas tipo" que enmarcan el desarrollo conceptual del curso, así como se explicita la transformación en el curso 1997 hacia el análisis del problema general de Ingeniería de Software de Sistemas de Tiempo Real.

¹ Inv. Principal CONICET. Profesor Tit. Ded. Excl., Dpto. de Informática, Facultad de Cs. Exactas, UNLP. E-mail degusti@info.unlp.edu.ar

² Prof. Adjunto con Ded. Excl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP. E-mail mnaiouf@info.unlp.edu.ar

³ Prof. Adjunto con Ded. Excl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP. E-mail laural@info.unlp.edu.ar

⁴ Calle 50 y 115 Primer Piso, (1900) La Plata, Argentina, Teléfono 54-21-227707. E-mail lidi@info.unlp.edu.ar

Introducción - Motivación

El tratamiento de datos en tiempo real ha sido uno de los ejes del desarrollo de la tecnología de hardware y software en los últimos 25 años. Aún antes del microprocesador, el control de procesos y máquinas industriales constituía una de las áreas de investigación tecnológica más dinámicas.

La evolución tecnológica y especialmente el tratamiento de señales (locales o remotas) ha impulsado enormemente esta área temática en los últimos años, sobre todo en los aspectos de concepción y desarrollo de software para Sistemas de Tiempo Real y Sistemas Distribuidos de Tiempo Real [Ben88] [Cou94] [Nie90] [Law92].

Los temas de Informática que subyacen son muy importantes: procesamiento distribuido, software de comunicaciones, procesamiento concurrente, ingeniería de software orientada a tiempo real (incluyendo arquitecturas, modelos, especificaciones, lenguajes, ambientes y algoritmos) [Bus88] [And91] [Geh91] [Set94].

En particular el valor agregado de los sistemas de software de tiempo real y sus áreas de aplicación (ej. control de vuelo, control de plantas industriales, monitoreo médico) hacen muy importante la verificación y validación de los sistemas de software, para lo cual es crítico disponer de especificaciones rigurosas y confiables [Geh86] [Lev90] [Shu92].

En este contexto resultó necesario incorporar la temática a la currícula normal de las carreras de Informática en la Universidad y en la UNLP desde 1989 se ha trabajado en un curso de Software de Sistemas de Tiempo Real [Inf95], cuyo enfoque y contenido son motivos del presente artículo.

Independientemente de la asignatura específica (que los alumnos pueden elegir en forma optativa en 4to. o 5to año), el contexto que ofrece la Licenciatura en Informática en la UNLP es favorable al tratamiento de los temas de tiempo real por 3 motivos:

1- Entre las asignaturas obligatorias el alumno tiene 2 cursos de Sistemas Operativos, 2 cursos de Ingeniería de Software, 2 cursos de Arquitectura de Procesadores, 1 curso de Redes y 1 curso de Programación Concurrente.

2- Hay un régimen muy flexible de optativas y entre ellas el alumno puede elegir varias relacionadas con la temática de Tiempo Real (Procesamiento de Señales, Sistemas Operativos Distribuidos, Sistemas Cliente-Servidor, Modelos y Simulación, Introducción a la Robótica, Procesamiento Paralelo, etc).

3- Existe una línea de investigación tradicional en el Departamento de Informática sobre Software de Tiempo Real, con acuerdos con el exterior y Doctorandos en el país y en el exterior, que aseguran una base de trabajo de investigación y desarrollo en el tema. [LIDI95].

Análisis de los enfoques posibles de un curso de Sistemas de Tiempo Real

El primer problema que encontramos al tratar de definir el enfoque y contenido del curso fue si lo centrábamos en alguno de los siguientes 5 aspectos:

- Arquitectura física de los sistemas de Tiempo Real. Análisis de Sistemas de Control y Sistemas de Consulta interactiva. Sistemas de Adquisición de Datos.
- Modelización y simulación de STR y SDTR.
- Especificación, verificación y validación de STR. Lenguajes formales de especificación.
- Lenguajes orientados a Tiempo Real y desarrollo de aplicaciones.
- Ingeniería de Software de Tiempo Real.

El segundo problema fue la relación entre Teoría y Práctica de estos temas y el peso relativo de cada una de ellas (lo que naturalmente era muy diferente según el enfoque elegido).

Naturalmente estas decisiones se componen de varios factores: el perfil del alumno que se está formando, la disponibilidad de docentes en el Dpto., las tendencias en el ámbito profesional en el país y en el exterior y los recursos físicos disponibles para implementar trabajos prácticos razonables.

En la realidad de los 5 enfoques posibles, hemos elegido aspectos de 3 últimos, por considerar que eran los más directamente relacionados con la formación de alumnos de Ciencia de la Computación. Durante estos años el peso relativo de estas 3 grandes temáticas ha variado (en relación con la oferta de asignaturas disponibles para los alumnos y su background previo).

Asimismo hemos establecido un conjunto de trabajos prácticos muy cercanos al control en Tiempo Real, a fin de que los alumnos desarrollen experiencias de interés pero sin darle otro significado que la motivación y la experimentación, es decir manteniendo el eje del curso en los temas teóricos de base que se cubren en la Teoría.

Naturalmente un alumno de Informática se siente interesado por los lenguajes y el desarrollo de aplicaciones, pero hemos tratado que el contenido teórico esencial se vincule con las especificaciones y la Ingeniería de Software, *porque la labor profesional de un especialista en Informática en el área de Tiempo Real es mucho más crítica en estos aspectos que en la codificación de algoritmos de control directo de hardware.*

De este conjunto de razonamientos parte la evolución de la asignatura, cuyos ejes temáticos y práctica actuales se analizan a continuación.

Ejes Temáticos del Curso

Podríamos definir los principales ejes temáticos del curso en los siguientes tres puntos:

1. Conceptos fundamentales de Tiempo Real, desde el punto de vista del Software. Incidencia de la distribución del procesamiento en la complejidad del ciclo de vida del software.
2. Modelos de especificación de la concurrencia en SDTR y su correlato en los lenguajes. Impacto de la distribución del procesamiento en el soporte de Sistema Operativo y de Comunicaciones.
3. Extensión de los conceptos y herramientas de Ingeniería de Software para el desarrollo de Sistemas de Tiempo Real.

La jerarquía de temas y objetivos siguió (aproximadamente) los siguientes lineamientos y pautas:

- 1- Definir y separar los componentes físicos de un Sistema de Tiempo Real (concentrado o distribuido). Ubicar los niveles de software y su complejidad relativa.
- 2- Analizar las características lógicas de los STR y SDTR y su impacto sobre el desarrollo de software. Poner especial énfasis en el aspecto tiempo *que relativiza la prueba funcional* de los sistemas, en el hecho de que en general un STR *no puede volver atrás* y en que muchas veces se está desarrollando un sistema *de tiempo infinito*.
- 3- Establecer los niveles de problema de Arquitectura, Comunicaciones, Sistema Operativo y Software (especificación, lenguajes, puesta a punto), marcando el objetivo de centrarse en los aspectos de la Ingeniería de Software de los STRs y STDRs.
- 4- Definir el problema de la concurrencia y discutir conceptualmente los mecanismos de memoria compartida y pasaje de mensajes, tratando que el alumno comprenda sus diferencias conceptuales y el impacto de los problemas de concurrencia en la especificación y verificación de STR y SDTR.
- 5- Discutir algunas herramientas de especificación formal, en particular CSP y Petri, orientando su utilización para describir fenómenos de Tiempo Real y su relación con los mecanismos explicados anteriormente.
- 6- Asociar los requerimientos para los lenguajes de programación concurrente con los modelos de solución en diferentes lenguajes. Vincular especificaciones con lenguajes. Explicar la derivación automática de código a partir de especificaciones.
- 7- Trabajar sobre todo el ciclo de vida de los STR y SDTR estableciendo las extensiones necesarias en la Ingeniería de Software clásica. Poner especial énfasis en la necesidad y requerimientos de herramientas CASE orientadas a Tiempo Real.

Evolución de los problemas planteados al alumno

Simultáneamente con el desarrollo de los temas teóricos, el alumno trabaja en un conjunto de prácticas experimentales cuya temática se puede resumir en:

- 1- Adquisición de datos y control.
- 2- Especificación.
- 3- Codificación de problemas concurrentes.
- 4- Desarrollo del ciclo de vida de un SDTR a partir de una metodología clásica extendida.
- 5- Utilización de herramientas CASE orientadas a Tiempo Real.

Limitaciones de las herramientas para la práctica

Una de las mayores preocupaciones fue el soporte de herramientas para la práctica. Además existe naturalmente un gap entre las herramientas teóricas de especificación y modelización de los problemas y las herramientas de implementación y verificación. En síntesis, la evolución investigó diferentes líneas:

1- Placas conversoras A/D y D/A

Se utilizaron dos modelos de placas conversoras: DAS-8 e INTELEKTRON. Las señales registradas eran obtenidas de :

- a) Diferentes generadores de señales analógicas, que permitían al alumno setear la forma de la onda (seno, cuadrada, triangular), la frecuencia y la amplitud de la señal.
- b) Sensor de temperatura.
- c) Micrófono

y procesadas de diferentes maneras:

- a) Gráficas
- b) Utilizando un hardware diseñado ad-hoc para la cátedra, comandado desde la puerta paralelo de manera que el alumno utilice las salidas digitales.
- c) La señal captada por el micrófono era posteriormente reproducida, utilizando la salida D/A, en un parlante.

Estas experiencias elementales, además de motivar al alumno, permiten apreciar las restricciones mínimas de un STR, desde diferentes puntos de vista. Aspectos como: exactitud en el procesamiento, imposibilidad de volver atrás, procesos de tiempo infinito, son claramente adquiridos utilizando estas herramientas.

2- Ambientes de especificación y verificación.

Se trabajó con diversas herramientas, especialmente basadas en CSP y en Petri. Se desarrolló un ambiente con Redes de Petri extendidas que permite especificar y simular STRs [Aba94] [Aba95].

3- Lenguajes de programación convencional.

Los alumnos trabajan en la práctica con lenguajes convencionales y ADA. También estudian OCCAM (por su relación directa con CSP) y pueden realizar trabajos especiales en OCCAM [Che84] [CSA90] [Inm87] [Usd81] [Hab83].

4- Herramientas de soporte para procesamiento distribuido.

Se trabajó con un soporte de comunicaciones distribuidas, basado en NetBios. Posteriormente se migró a PVM y actualmente se está estudiando MPI [Lei92] [Mor94] [Pvm96]

En todos los casos este soporte lógico permite utilizar una red heterogénea como arquitectura MIMD para la prueba de algoritmos distribuidos.

5- Herramientas CASE extendidas para Tiempo Real

Se desarrolló un ambiente CASE (ISTRAC) orientado a Tiempo Real, que es utilizado por los alumnos y permite seguir una metodología estructurada extendida, similar a los conceptos que se dan en Teoría [Mar95].

6- Trabajos en lápiz y papel sobre modelos y problemas teóricos.

Naturalmente se han mantenido la especificación de problemas teóricos y su verificación/validación formal [Pet81] [Roz88] [Hoa85].

Qué se investiga y desarrolla con alumnos en Tiempo Real ?

Naturalmente la temática de Tiempo Real es muy amplia y excede lo que se puede cubrir en el curso regular. De todos modos el trabajo realizado con alumnos (ya sea como trabajos finales de la asignatura como Trabajos de Tesina de Licenciatura) ha dado muchos resultados interesantes:

- Estudio de diferentes herramientas de especificación y validación formal. (Estelle, CCS, QPNet, LDHs)
- Desarrollo de derivadores de código a partir de especificaciones formales (en particular CSP-ADA y ESTELLE-ADA).
- Desarrollo de herramientas orientadas a la especificación y verificación con Redes de Petri extendidas.
- Desarrollo de herramientas CASE orientadas a algún aspecto del ciclo de vida de software.
- Estudio de problemas derivados de la distribución de los datos, tales como consistencia en Bases de Datos distribuidas en sistemas sujetos a restricciones de tiempo.
- Modelización de subsistemas de comunicaciones y protocolos en ambientes distribuidos de tiempo real, a fin de estudiar su performance.
- Desarrollos de controladores dedicados específicos, con control directo de hardware (especialmente en aplicaciones médicas con la Facultad de Medicina).

Resultados Obtenidos y Líneas de Trabajo actuales

Los resultados de un curso universitario son muy difíciles de evaluar, o al menos tienen diferentes pautas para su evaluación. De todos modos, trataremos de sintetizar algunos resultados:

- Al tratarse de una asignatura optativa en la cual el perfil de interés de los alumnos es alto hemos tenido muy buena respuesta en el aprendizaje y la investigación de temas por los alumnos.
- Los contenidos y la bibliografía requieren un análisis permanente por la vigencia actual del tema. De todos modos está claro que para los alumnos de la UNLP la base de arquitecturas y redes la tienen muy bien cubierta, por lo que pusimos especial énfasis en los temas de especificaciones, lenguajes, derivación automática de código e Ingeniería de Software de tiempo real.
- Hemos puesto en marcha una serie de herramientas accesibles por los alumnos (Placas A/D y D/A, NetBios, PVM, Redes de Petri extendidas, CASE para Tiempo Real, Compiladores OCCAM, C paralelo y ADA).
- El sustento de un proyecto de investigación existente en el Departamento de Informática (referido a Software de Sistemas de Tiempo Real) ha permitido orientar varios Trabajos de Grado de alumnos en esta línea.
- La línea de trabajo actual supone la incorporación de temas de Redes Neuronales al tratamiento de datos y decisiones en Tiempo Real. Si bien el tema es motivo de otra asignatura optativa, algunos de los conceptos y desarrollos se incorporarán a los trabajos de alumnos de Tiempo Real. Por otra parte se desea potenciar el ambiente CASE ISTRAC a fin de mejorar su utilización por los alumnos.

Conclusiones

Se ha presentado la implementación de un curso de Software de Sistemas de Tiempo Real como asignatura optativa de la Licenciatura en Informática de la UNLP, poniendo énfasis en la secuencia de decisiones respecto de contenidos, enfoque temático y recursos a utilizar que los autores fueron tomando desde 1989.

La experiencia ha sido muy positiva y coherente con las tendencias crecientes de tratamiento de señales y datos en tiempo real, permitiendo ofrecer a los alumnos una opción de iniciarse en el tema.

Bibliografía

- [Aba94] Abásolo, Cantarella, Naiouf, Pesado, De Giusti "Especificación y derivación de sistemas de Tiempo Real con Redes de Petri extendidas". Proceedings del I International Conference on Information Engineering. Buenos Aires. 1994.
- [Aba95] Abásolo, Cantarella, De Giusti "Redes de Petri extendidas. Derivación de Código ADA". Trabajo de Tesina de Ingeniero en Sistemas UNCPBA. Expuesta en 1995.
- [And91] Andrews, "Concurrent Programming", Benjamin/Cummings, 1991.
- [Ben88] S. Bennet, "Real-Time Computer Control: An Introduction", Prentice Hall, 1988.
- [Bus88] Bustard, Elder, Welsh, "Concurrent Program Structures", Prentice Hall, 1988.
- [Che84] Cherry, "Parallel programming in ANSI standard ADA", Prentice Hall, 1984.
- [Cou94] G. Coulouris, J. Dollimore, T. Kindberg, "Distributed Systems. Concepts and Design", Addison Wesley, 1994.
- [CSA90] "OCCAM", Computer System Architects, 1990.
- [Geh86] N. Gehani, A. D. McGettrick, "Software Specification Techniques", Addison Wesley, 1986.
- [Ghe91] Ghezzi, Jozayeri, "Programming lenguajes concepts", John Wiley, 1991.
- [Hab83] Habermann, Perry, "ADA for experienced programmers", Addison Wesley, 1983.
- [Hoa85] C. A. R. Hoare, "Communicating Sequential Processes", Pentice-Hall, 1985.
- [Inf95] Dpto. de Informática. Plan de Estudios vigente 1995-96
- [Inm87] Inmos, "The OCCAM Programming Language", Inmos, 1987.
- [Law92] H. Lawson, "Parallel processing in industrial real time applications", Prentice Hall 1992.
- [Lei92] F. T. Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 1992.
- [Lev90] Shem-Tov LEVI, Ashok Agrawala, "Real Time System Design", McGraw-Hill Inc, 1990.
- [Mar95] M. Mariangelli, J. Spognardi, P. Pesado, A. De Giusti, "Diseño e Implementación de un Ambiente CASE para Sistemas de Tiempo Real", Proceedings del I CACIC. Bahía Blanca, pp340-352, 1995.
- [Mor94] H. S. Morse, "Practical Parallel Computing", AP Professional, 1994.
- [Nie90] Kjell Nielsen, "Ada in Distributed Real-Time Systems", McGraw-Hill, 1990.
- [Pet81] J. Peterson, "Petri Nets Theory and the Modeling of Systems", Prentice-Hall, 1981.
- [PVM96] "Parallel Virtual Machine", World Wide Web.
- [Roz88] Rozenberg (Ed.), "Advances in Petri Nets", Springer-Verlag, 1988.
- [Set94] Sethi, "Lenguajes de Programación. Conceptos y Constructores", Addison-Wesley, 1994.
- [Shu92] K. Shumate, M. Keller, "Software Specification and Design", John Wiley & Sons, 1992.
- [Usd81] United States Department of Defense, "The programming language ADA - Reference Manual". Springer Verlag. 1981