

Herramienta interactiva para la enseñanza temprana de Concurrencia y Paralelismo: un caso de estudio.

Laura De Giusti¹, Fabiana Leibovich¹, Mariano Sánchez¹, Sebastian Rodriguez Eguren¹, Franco Chichizola¹, Marcelo Naiouf¹, Armando De Giusti^{1,2}

¹ Instituto de Investigación en Informática LIDI (III-LIDI) – Facultad de Informática –UNLP

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Argentina

{ldgiusti, fleibovich, msanchez, seguren, francoch, mnaiouf, degiusti}@lidi.info.unlp.edu.ar

Abstract. Se analiza el impacto del uso de una herramienta interactiva para la enseñanza de conceptos de concurrencia en un curso CS1. Para ello se toma como caso de estudio 40 alumnos divididos en 14 grupos, que utilizando el entorno CMRE, implementan soluciones a problemas clásicos de concurrencia para su trabajo final de fin de cursada. Además se ejemplifica una experiencia en el uso de robots físicos que ejecutan en tiempo real las mismas instrucciones que los robots virtuales y se comunican con el entorno mediante bluetooth.

Keywords: Concurrencia, Paralelismo, Currícula, Entorno, Multirobot, Algoritmos Concurrentes y Paralelos.

1 Introducción

La utilización de herramientas interactivas para la enseñanza de conceptos fundamentales a alumnos de un curso CS1 [ACM04][ACM08][ACM13](Algoritmos, Datos y Programas en nuestro caso) se ha vuelto una práctica generalizada, considerando los aspectos motivacionales y las actividades habituales de los jóvenes que ingresan a la Universidad: normalmente viven rodeados de pantallas táctiles, celulares inteligentes, tablets, etc y su interacción con las mismas es permanente [DEG13]. En este contexto, proveerles un entorno gráfico e interactivo permite reducir la brecha que tradicionalmente existió entre la abstracción y la posibilidad de ver gráficamente la aplicación de los conceptos estudiados [AMD09][HOO13].

2 Objetivos del entorno multirrobot en desarrollo

En este trabajo se presenta una experiencia práctica de la utilización del entorno CMRE [DEG14][DEG14b] en alumnos de promoción de un curso CS1 (Algoritmos, Datos y Programas – curso anual de primer año en las Licenciaturas en Informática y Sistemas de la UNLP). Para ello se tomó un subconjunto de alumnos, en comisiones de dos y tres alumnos, y se les entregó a cada uno un enunciado diferente para que pudieran resolver el problema correspondiente utilizando múltiples robots en el entorno. Los problemas elegidos se basan en los típicos problemas de concurrencia que se abordan en la bibliografía clásica para entender y explicar los conceptos básicos [AND00].

2.1 Resumen de las características del Entorno CMRE

Tal como se mencionó anteriormente pueden resumirse las capacidades del ambiente CMRE de la siguiente manera [DEG12a][DEG12b][DEG12c][GEP06][DEG14]:

- Existen múltiples procesadores (robots) que realizan tareas y que pueden cooperar y/o competir. Los mismos representan los “cores” de una arquitectura multiprocesador real. Estos robots virtuales pueden tener un reloj propio y diferentes tiempos para la ejecución de sus tareas específicas.
- El modelo de ambiente (“ciudad”) en la que desarrollan sus tareas admite áreas privadas, parcialmente compartidas y totalmente compartidas. En una área privada sólo puede moverse un único robot, en un área parcialmente compartida se especifica el conjunto de robots que pueden moverse en ella y en un área totalmente compartida todos los robots definidos en el programa pueden moverse dentro de ella.
- Si se instancia a un sólo robot en un área que abarque toda la ciudad, se repite el esquema del Visual Da Vinci.
- Cuando dos o más robots están en un área compartida (parcial o totalmente), compiten por el acceso a las esquinas del recorrido y a los recursos que allí existan. Para esto deben sincronizar.
- Cuando dos o más robots (en un área común o no) desean intercambiar información (datos o control) deben hacerlo por mensajes explícitos.
- La sincronización se da por un mecanismo equivalente a un semáforo binario.
- La exclusión mutua puede generarse con la declaración de las áreas alcanzadas por cada robot. Acceder a otras áreas de la ciudad, así como salir de ellas no está permitido.
- Todo el modelo de ejecución es sincrónico y permite la existencia de un reloj virtual de ciclos, que a su vez permite asignar tiempos específicos a las operaciones, simulando la existencia de una arquitectura heterogénea.
- El entorno permite ejecutar el programa de manera tradicional, o paso a paso por instrucciones, dando al usuario un control detallado sobre la ejecución del programa, de manera de poder controlar situaciones típicas de concurrencia tales como conflictos (colisiones) o deadlocks.
- En la ejecución paso a paso, el efecto de las operaciones se puede reflejar en los

robots físicos, comunicados vía Bluetooth.

3 Enunciados elegidos y experiencias reales

Acá describir algunos ejemplos de enunciados, describir las formas de implementación, problemas encontrados, correspondencia del entorno con ejemplos prácticos de concurrencia propiamente dicho. Describir tiempos de desarrollo, cantidad de grupos empleados, recomendaciones de los mismos.

Tema A

En este problema pondremos una zona común para todos los procesos que va desde la esquina (15,2) a la esquina (20,3). En esta zona cada uno de los seis procesos robotLector querrá tomar una flor y cada uno de los 3 procesos robotEscritor querrá depositar una flor de la siguiente manera:

Cada robot lector se posicionará en una esquina dentro del área compartida cuando la misma esté disponible para poder tomar una flor. Mientras tanto cada proceso escritor colocará una flor en cada esquina del área compartida, cuando esto sea posible. Hay que tener en cuenta que puede haber múltiples lectores en diferentes esquinas del área compartida pero en el caso de los escritores, solo podrá haber uno de ellos en el área compartida la cual no deberá tener ni lectores ni otros escritores.

Robot Organizador: debe coordinar el acceso al área compartida por los lectores y escritores.

Para terminar el programa cada robot debe haber realizado su trabajo 3 veces.

Puede asegurarse que todos los procesos tengan flores para depositar y que haya flores en las esquinas.

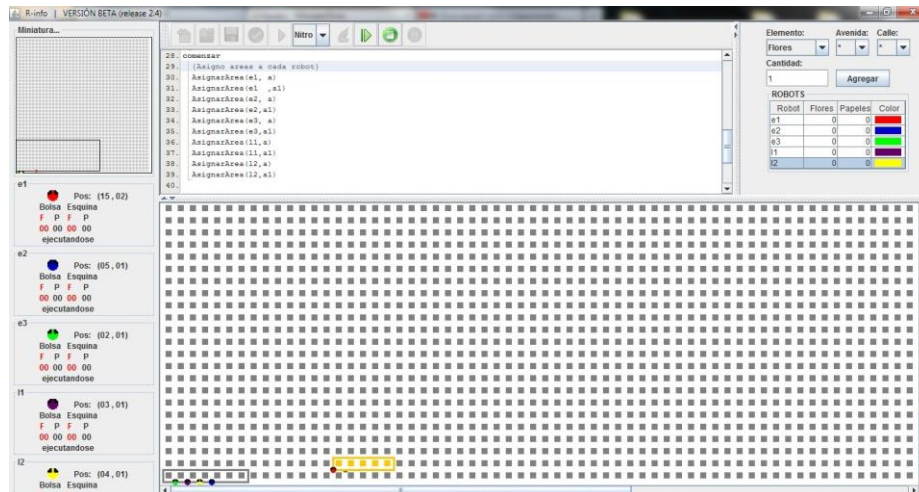


Fig. 1 Tema A

El enunciado representa el clásico problema de “Lectores Escritores”: un conjunto de procesos lectores que quieren leer datos de una BD y un conjunto de escritores que quieren escribir en la Base de Datos (BD). Para esto varios lectores pueden leer al mismo tiempo pero sólo un escritor puede escribir en la BD (tampoco puede haber lectores mientras el escritor está en la BD). [MCC08]

Tema B

Existen 8 robots que deben “limpiar de papeles su área”. Para esto deben juntar todos los papeles que se encuentran en su área privada.

La tarea colaborativa de los robots se realiza de la siguiente manera:

Cada robot tendrá un área privada, la cual deberá limpiar de papeles.

Robot Organizador: comienza el juego ubicado en la esquina (1,1). Luego debe inicializar cada robot en una esquina al azar dentro de su área (mirando hacia arriba). Al finalizar debe informar la cantidad de papeles juntados entre todos los robots.

Cuando el robotOrganizador le envía la esquina (avenida,calle) por la que cada robot debe comenzar el recorrido.

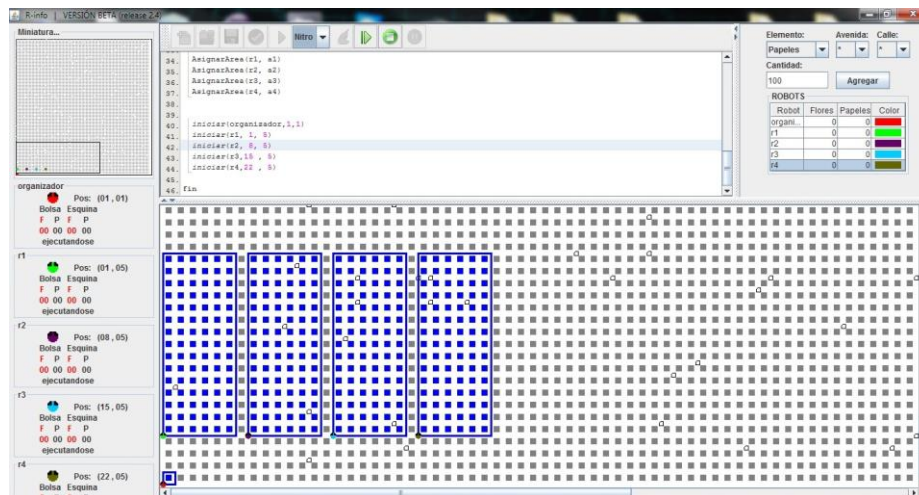


Fig. 2 Tema B

Este problema representa un típico ejercicio de memoria distribuida utilizando el patrón de interacción master/worker. Cada proceso worker posee su propio espacio de direcciones, y ejecutan el mismo código (SIMD). Al finalizar, envían la información a un proceso master que se encargará de totalizarla. Es importante notar que la comunicación es por mensajes, con primitivas provistas por el entorno CMRE.

Tema C

Supongamos que en la esquina (20,20) de la ciudad hay una cantidad al azar de flores. Además se tienen 4 robots Tomadores que compiten entre ellos para ver quien junta más flores de la esquina (20,20) de la siguiente manera:

robotTomadorX: comienza en la esquina (i,j). Para saber si puede juntar una flor debe tirar un número al azar entre 0 y 1. Si el número es:

- 0 debe quedarse donde está sin hacer nada.
- 1 espera a que no haya otro robot en la esquina (20,20) y en caso de haber toma 1 flor de esa esquina. En el caso de que no haya más flores debe terminar e informarle al robotJefe cuantas flores ha juntado.

robotJefe: comienza en la esquina (1,1). Debe determinar que robotTomador ha ganado.

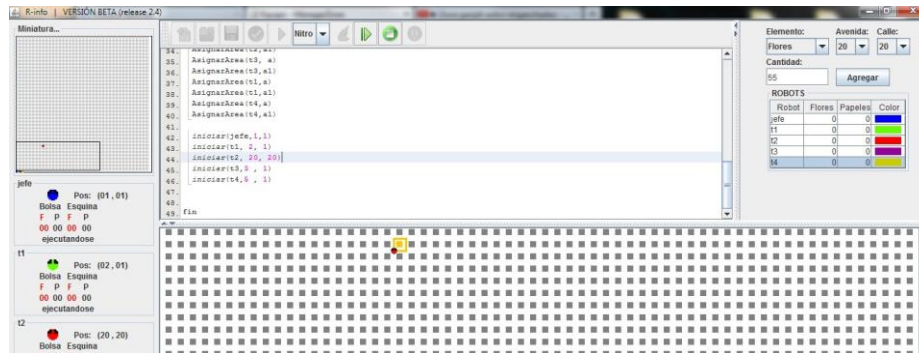


Fig. 3 Tema C

Este enunciado representa el típico problema de exclusión mutua sobre memoria compartida. Por otra parte, también se utiliza el mecanismo de comunicación por mensajes, con el proceso que determina el robot “ganador”.

4 Utilización de robots físicos

Lego Mindstorms EV3 es el robot de tercera generación en LEGO 's Mindstorms línea de la robótica [LEGO][LEGOb].

El EV3 tiene un procesador basado ARM9 el cual corre en Linux (Debian 6), LCD monocromo 178 x 128 píxeles, Memoria principal 64 MB de RAM ,16 MB Flash. Un conector USB y una ranura Micro SD.

¿Qué es leJOS? leJOS es una pequeña máquina virtual de Java.

leJOS para el EV3 incluye todas las clases de la API EV3 y un sistema Java run time.

leJOS EV3 ofrece lo siguiente:

- Lenguaje orientado a objetos (Java)
- Hilos de Preferencia (tareas)
- Arrays, incluyendo multidimensionales
- Recursión
- Sincronización
- Excepciones
- Tipos Java incluyendo float, long y String

- La mayoría de las clases `java.lang`, `java.util` y `java.io`
- Un API Robótica bien documentada

Es importante aclarar que la utilización de los robots físicos se utilizan con fines ilustrativos y como atractivo e incentivador para los alumnos.

En el último congreso TEYET [Referencia a Teyet] se realizó un demo utilizando el robot lego, descrito previamente, conectado a la pc donde se ejecutaba el entorno mediante bluetooth. El robot crea un Server socket en un puerto determinado, al cual el entorno se conecta y envía las instrucciones a realizar secuencialmente (Paso a paso). El robot físico está siempre a la espera de las instrucciones enviadas desde el entorno. Cuando la recibe, decodifica la misma y la ejecuta. Una vez finalizada le avisa al entorno que terminó para que este le pueda enviar otra instrucción a realizar. De esta manera el entorno y el robot trabajan de manera sincrónica.

5 Resultados obtenidos

Tal como se especificó en la sección 2, 14 grupos de entre 2 y 3 alumnos realizaron su trabajo final utilizando el entorno CMRE.

Se les plantearon 14 problemas diferentes, todos con eje en conceptos básicos de concurrencia. En este paper se mencionan 3 de ellos.

Todos los grupos culminaron su trabajo final satisfactoriamente con nota superior a 8, respetando el plazo de un mes y medio para la entrega desde la recepción del enunciado. Posteriormente defendieron en un coloquio las soluciones desarrolladas, además de verificar su corrección.

Entrevistas a los alumnos mostraron entusiasmo por la utilización de la herramienta y también por la resolución de problemas de concurrencia pudiendo aplicar los conceptos aprendidos durante la cursada (estructuras de control, tipos de datos, modularización, pasaje de parámetros) así como los conceptos de utilización de múltiples robots sobre recursos compartidos, con acceso a memoria compartida y pasaje de mensajes y con una efectiva implementación de los conceptos de comunicación y sincronización entre procesos. .

Por otro lado, los alumnos propusieron posibles modificaciones y funcionalidades a agregar al entorno, que son tenidas en cuenta mientras se continúa en el desarrollo de mejoras en el mismo.

6 Conclusiones y Líneas de Trabajo Futuro

Se ha presentado un caso de estudio realizado con alumnos de primer año, con el objetivo de analizar el aprendizaje de conceptos básicos de concurrencia y paralelismo en un curso CS1.

El empleo del entorno CMRE como herramienta para favorecer la prueba interactiva de algoritmos concurrentes ha sido muy positivo y su combinación con robots físicos para demos para los alumnos resulta motivante para los alumnos.

En 2014 y 2015 la experiencia se extenderá a todos los alumnos de los cursos CS1 de las Licenciaturas y de la Ingeniería en Computación, trabajando al mismo tiempo diferentes perfeccionamientos/extensiones del entorno CMRE y su empleo en otras asignaturas.

7 Bibliografía

- [ACM04] ACM/IEEE-CS Joint Task Force on Computing Curricula. "Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering". Report in the Computing Curricula Series. 2004.
- [ACM08] ACM/IEEE-CS Joint Interim Review Task Force. "Computer Science Curriculum 2008: An Interim Revision of CS 2001". Report from the Interim Review Task Force. 2008.
- [ACM13] ACM/IEEE-CS Joint Task Force on Computing Curricula. "Computer Science Curricula 2013". Report from the Task Force. 2013.
- [AMD09] AMD. "Evolución de la tecnología de múltiple núcleo". <http://multicore.amd.com/es-ES/AMD-Multi-Core/resources/Technology-Evolution>. 2009.
- [AND00] Andrews G. "Foundations of Multithreaded, Parallel, and Distributed Programming". Addison Wrsley, 2000.
- [CAR03] Carr S., Mayo J., Shene C. "Threadmentor: a pedagogical tool for multithreaded programming". ACM Journal of Educational Resources, 3:1–30, 2003.
- [DEG12a] De Giusti A. E., Frati F. E., Leibovich F., Sánchez M., De Giusti L. C., Madoz M. C. "Concurrencia y Paralelismo en CS1: la utilización de un Lenguaje Visual orientado". Proceeding del VII Congreso de Tecnología en Educación y Educación en Tecnología. 2012
- [DEG12b] De Giusti L. C., Frati F. E., Leibovich F., Sánchez M., Madoz M. C. "LMRE: Un entorno multiprocesador para la enseñanza de conceptos de concurrencia en un curso CS1". Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología. Págs. 7 - 15. 2012.
- [DEG12c] De Giusti A. E., Frati F. E., Sánchez M., De Giusti L. C. "LIDI Multi Robot Environment: Support software for concurrency learning in CS1". Proceeding of IEEE International Conference on Collaboration Technologies and Systems. Pág. 294-298. 2012.
- [DEG13] De Giusti A. E. "El cambio tecnológico como motor de la Investigación en Informática". Conferencia inaugural del Workshop de Investigadores en Ciencia de la Computación (WICC2013). 2013.
- [GEP06] Gepner P., Kowalik M.F. "Multi-Core Processors: New Way to Achieve High System Performance". In: Proceeding of International Symposium on Parallel Computing in Electrical Engineering 2006 (PAR ELEC 2006). Pags. 9-13. 2006.
- [HOO13] Hoonlor A., Szymanski B. K., Zaki M. J., Thompson J. "An Evolution of Computer Science Research". Communications of the ACM. 2013.
- [LEGO] "Lego Education". <http://www.legoeducation.us/eng/characteristics/ProductLine~LEGO%20MINDSTORMS%20Education%20EV3>.
- [LEGOB]Lego. "LEGO Mindstorms EV3 Announced". <http://brickextra.com/2013/01/10/lego-mindstorms-ev3-announced/>
- [MCC08] McCool M. "Scalable Programming Models for Massively Parallel Multicores". Proceedings of the IEEE, 96(5): 816–831, 2008.

- [DEG14] De Giusti, L., Leibovich, F., Sanchez, M., Chichizola, F., Naiouf, M., De Giusti, A. “Desafíos y herramientas para la enseñanza temprana de Concurrencia y Paralelismo”. Congreso Argentino de Ciencias de la Computación (CACIC), 2014.
- [DEG14b] De Giusti, A., De Giusti L., Leibovich, F., Sanchez, M., Rodriguez Eguren, S. “Entorno interactivo multirrobot para el aprendizaje de conceptos de Concurrencia y Paralelismo”. Congreso Tecnología en Educación, Educación en Tecnología. 2014.