



Experiencias con la enseñanza de programación en ambientes colaborativos

Gustavo de la Cruz Martínez y Fernando Gamboa Rodríguez
Centro de Ciencias Aplicadas y Desarrollo Tecnológico
Universidad Nacional Autónoma de México

Resumen

La necesidad de la enseñanza de los lenguajes de programación se extiende cada día más a diferentes áreas, incluso en aquellas consideradas alejadas de la ingeniería de software. La mayoría de los métodos de enseñanza fueron diseñados para personas con cierto perfil, y aunque se ha buscado diseñar lenguajes de programación más “simples” aún las diferentes metodologías de programación generan un sesgo en la población que puede utilizarlas. Así, ante la necesidad de enseñar a programar a personas cuyo perfil no presente características afines con el desarrollo de software, planteamos los resultados obtenidos al enseñar a programar con un lenguaje de programación basado en java, pero con las ventajas de usar los mecanismos propuestos para los lenguajes de programación visual.

Introducción

El desarrollo de programas es una actividad cognitiva de alto nivel y a decir de diferentes investigadores requiere desarrollar representaciones abstractas en forma de estructuras lógicas. Los modelos mentales (también referidos como esquemas) juegan un papel importante en la comprensión de programas, (Pennington, 1987; Soloway, 1984; Wiedenbeck et al 1999) así como en las tareas de comprensión relacionadas, modificación y corrección de errores.

Se considera que desarrollar programas es básicamente una actividad de solución de problemas, como señala Greenfield (1987) la habilidad de resolver problemas no se adquiere por la práctica de “resolver problemas” sino que las actividades deben ser implementadas y planeadas cuidadosamente para asegurar la transferencia en los alumnos.

Actualmente, en un curso de ingeniería de software a nivel de licenciatura, el maestro cuenta con diferentes herramientas CASE que facilitan la puesta en práctica de métodos de concepción y de lenguajes de programación. Sin embargo, la experiencia muestra que para un alumno de este nivel de estudios, el problema para desarrollar el software de un sistema no reside en dominar una notación de una metodología o la sintaxis de un lenguaje de programación, sino el problema reside en adquirir experiencia en plantear tanto el problema y la propuesta de una solución que permita pasar después al desarrollo del software.

Si consideramos la enseñanza de otras áreas, como la inteligencia artificial, que requieran del uso de lenguajes de programación para poner en práctica los conocimientos revisados, el planteamiento y la solución se vuelven el principal objetivo,



pero si el estudiante no conoce el lenguaje que será utilizado no es fácil obtener buenos resultados.

Así nos encontramos ante una gran cuestión: cómo enseñar inteligencia artificial, por ejemplo, a un grupo de alumnos no expertos en programación, sin tener que enseñarles todos los detalles de metodología, semántica y sintaxis de un lenguaje de alto nivel. En este trabajo describimos algunas de las experiencias que obtuvimos al plantear el curso de inteligencia artificial usando un lenguaje visual, que permita a los estudiantes poner en práctica sus conocimientos sin tener que ser un experto en un lenguaje de programación de alto nivel como lo es java.

Enseñando Inteligencia Artificial

Durante mucho tiempo la enseñanza de la Inteligencia Artificial ha estado dirigida a estudiantes del área de cómputo o estudiantes con ciertos conocimientos en programación; por esta razón, el profesor asume que sus estudiantes son capaces de desarrollar los programas de cómputo que les permitan resolver ciertos problemas, haciendo uso de las técnicas de solución de problemas asociadas a la Inteligencia Artificial. Pero, ¿qué pasa cuando esta suposición no es totalmente cierta, es decir, cuando los estudiantes no son especialistas en cómputo o en el desarrollo de software? En estos casos los estudiantes se ven limitados al poner en práctica los conceptos que se les presenta en clase y al final pueden tener una visión limitada de lo que realmente es la Inteligencia Artificial.

Con el fin de buscar herramientas que apoyen a los estudiantes en el aprendizaje de la Inteligencia Artificial, se han desarrollado de una serie de herramientas de software que faciliten la actividad de desarrollar programas y les permitan poner en práctica sus conocimientos acerca de la Inteligencia Artificial. En la primera fase del desarrollo de una de estas herramientas, *Puzzle*, considera algunos conceptos básicos de la Inteligencia Artificial relacionados con el comportamiento, el manejo del conocimiento y su uso.

Puzzle es una herramienta que permite al usuario construir programas de cómputo en un ambiente colaborativo, siguiendo la filosofía de la programación visual. Cada estudiante puede colaborar en la implementación de un algoritmo, sin la necesidad de tener grandes conocimientos acerca de la sintaxis de un lenguaje de alto nivel, y aplicar dicho algoritmo a la resolución de problemas.

Primeras observaciones

Hemos visto que con *puzzle* se estimula la capacidad crítica y de propuesta de los alumnos, al tener que dar soluciones al problema establecido y la necesidad de cubrir un objetivo.



Se ha observado que utilizando el ambiente que se proporciona con esta herramienta el estudiante puede, de manera más simple, poner en práctica los conceptos básicos que se presentan en un curso de IA. La gran ventaja que se le da al estudiante es que ahora no se preocupa por los detalles de la implementación de los diferentes algoritmos y técnicas, ni tampoco necesita ser un experto en el manejo de algún lenguaje de programación de alto nivel. De esta manera se puede comprobar de manera simple las ventajas y limitaciones de los diferentes métodos.

Una aportación importante de *puzzle* es que ayudó al docente a abarcar de manera más completa el plan de estudios, permitiéndole adentrarse más en las áreas de aplicación, describiendo los trabajos actuales en dichas campos.

Experiencias obtenidas

Creemos que el uso de herramientas que permitan poner en práctica los conceptos teóricos ayuda al estudiante a entenderlos de manera más adecuada, ya que le sirve para percibir las ventajas y limitaciones de los diferentes conceptos que se le presentan. Esto es particularmente importante en cursos de introducción, por la brevedad de los cursos y lo extenso del temario.

En particular con la enseñanza de la Inteligencia Artificial es primordial atraer al estudiante y aclarar su concepción de ella, para mostrar el alcance real del área.

El ahorro de tiempo en la implementación de los algoritmos permite además que los alumnos tengan un mayor margen para poder investigar y profundizar por si mismos algunas alternativas no propuestas o comentadas en clase.

Esto también es una muestra que disipa otro mito muy arraigado sobre la IA y corresponde a que solo se deben o pueden hacer desarrollos de esta área en lenguajes de programación como LISP o PROLOG.

Bibliografía

Campione, M., Walrath, K., Huml, A. The Java Tutorial, Sun. 3a. ed. EUA. 2001

Jain, L. C. et. al. Virtual Environments for Teaching & Learning. World Scientific. Londres. 2005.

Rich, E. y Knight, K. Artificial Intelligence. McGraw-Hill. 2a. ed. EUA. 1991.

Russell, S. y Norving, P. Artificial Intelligence a modern approach. Prentice Hall. EUA 1996.

Santos, J. y Duro, R. Evolución Artificial y Robótica Autónoma. RA-MA. Madrid. 2005.