

Plataforma de enseñanza de lenguajes de programación a través de Internet: Proyecto IDEFIX

Jose E. Labra Gayo, José M. Morales Gil
Dpto. de Informática
Universidad de Oviedo, España
e-mail: labra@lsi.uniovi.es jmmoral@correo.uniovi.es

Roberto Turrado C.
Fachhochschule of Ulm
Alemania
e-mail: rturrado@yahoo.com

Resumen

En este artículo se describe la arquitectura del proyecto IDEFIX cuyo objetivo es desarrollar una plataforma que facilite la enseñanza de la programación en diferentes lenguajes mediante la utilización de Internet. El sistema permite la realización de prácticas de laboratorio mediante la creación de un entorno dinámico de desarrollo basado en Internet. En este entorno, los estudiantes tienen acceso a través de Internet a los enunciados de los ejercicios de programación escritos en un formato XML, que facilita la presentación en sistemas heterogéneos y que permite la posterior evaluación de forma automática. El sistema facilitará la realización interactiva de los ejercicios monitorizando los resultados parciales, fomentando el desarrollo colaborativo y facilitando la automatización del proceso de evaluación.

1. Introducción.

La Universidad de Oviedo, junto con otras seis universidades españolas del denominado G7 participa en la creación de un campus virtual que permita la enseñanza compartida de asignaturas de libre elección a través de Internet. Dentro de dicho campus, el proyecto AulaNet [17], propio de la Universidad de Oviedo, ha incorporado la asignatura “Programación lógica y funcional” desde el curso 2001/2002. Dicha asignatura es impartida por uno de los autores de este artículo y consiste principalmente en la presentación de los paradigmas de programación lógico y funcional con sus correspondientes prácticas en los lenguajes Prolog y Haskell.

Existen en el mercado diversas herramientas que facilitan la publicación de cursos generales en

Internet. En particular, en el proyecto AulaNet se utiliza el sistema WebCT como herramienta común básica de desarrollo. Sin embargo, la enseñanza de lenguajes de programación tiene ciertas particularidades que suponen un mayor reto para su implementación en la red.

El proyecto IDEFIX (*Integrated Development Environment Frameworks based on Internet and eXtensible technologies*) [12] se centra en el desarrollo de entornos integrados de desarrollo a través de Internet. El proyecto está formado por investigadores del grupo Oviedo3 de la Universidad de Oviedo y de la Fachhochschule of Ulm, para la cual se está desarrollando un sistema de realización de prácticas de laboratorio a través de Internet de la asignatura de Sistemas Distribuidos. En este artículo se describe la justificación y las intenciones que persigue dicho proyecto.

2. Antecedentes.

Dentro de las principales funciones que cumplen los profesores a cargo de los cursos de lenguajes de programación, está la supervisión de las prácticas de laboratorio, en donde debe encargarse de revisar y asesorar al alumno, sobre los problemas que se le plantean al llevar a la práctica los conocimientos adquiridos en las clases teóricas.

La realización tradicional de prácticas de laboratorio tiene una serie de inconvenientes, entre los que se pueden destacar [9].

- En la mayoría de los casos, el profesor no cuenta con un tiempo suficiente para atender a la totalidad de los alumnos de manera particular y resolver cada una de sus dudas.
- En muchos centros, no se cuentan con instalaciones suficientes para satisfacer la

- demanda de equipos necesarios para llevar a cabo las prácticas.
- El ejercicio de las prácticas, se efectúa tiempo después de haber obtenido el conocimiento teórico, lo que presupone un esfuerzo mayor para refrescar los conocimientos provocando el surgimiento de nuevas dudas.
- La posibilidad de detectar plagios, se torna una tarea muy complicada que reclama mucho tiempo del maestro.
- Los maestros dedican mucho tiempo a tareas rutinarias, que les impiden desarrollar actividades más creativas.

La realización de estas prácticas de laboratorio a través de Internet podría subsanar algunos de los puntos anteriores.

3. Ventajas e inconvenientes de la educación a distancia.

Mucho se habla de los beneficios que las nuevas tecnologías están aportando en todos los campos, y si se considera específicamente la educación, se podría decir que, si bien resultan innegables los grandes logros y aportaciones que en este campo se están dando, también sería justo mencionar que se han creado los siguientes mitos, que amenazan con afianzarse sin su correspondiente proceso reflexivo ni empírico [6].

- La tecnología intensifica la enseñanza y el aprendizaje
- Los ordenadores han cambiado la forma de trabajar
- Los ordenadores han cambiado la forma de enseñar
- La introducción de tecnología en el proceso de enseñanza es innovadora
- Los alumnos aprenden más fácilmente con ordenadores
- La tecnología solucionará los conceptos erróneos de los estudiantes

Lo anterior nos obliga a realizar un análisis más cuidadoso acerca de los beneficios e inconvenientes que representa el hacer uso de Internet para propósitos educativos, evitando hacerlo de forma indiscriminada y atendiendo a sus repercusiones. Varios autores [2,4] dan cita a los aspectos favorables del empleo de la red en forma adecuada, destacando los siguientes:

- Evita los desplazamientos de los usuarios.
- Flexibiliza los horarios educativos.
- Promueve la formación de hábitos.
- Individualiza el ritmo de aprendizaje.
- Diversifica los métodos y las tecnologías educativas.
- Facilita la actualización de los contenidos.
- Permite la simulación de actividades.
- Facilita el acceso y aumenta las referencias bibliográficas.
- Promueve la participación activa y el trabajo colaborativo.
- Facilita el trabajo administrativo del maestro.
- Retroalimenta al maestro y le permite reorientar el curso oportunamente.

Obviamente, las mismas fuentes citan los siguientes inconvenientes:

- Escasez de relaciones humanas.
- Resistencia al cambio de parte del profesorado.
- Necesidad de capacitar a los maestros.
- La legitimidad de los procesos de evaluación no está lo suficientemente garantizada.
- El prestigio de los títulos obtenidos por este medio, no gozan de mucho reconocimiento.
- La libertad y flexibilidad pueden resultar contraproducentes para los alumnos inmaduros.
- Incertidumbre sobre la eficacia de los resultados y sus efectos psicológicos.
- Los problemas técnicos y los costes que trae consigo el uso de Internet como plataforma de trabajo.

Lo anterior, invita a reflexionar sobre el uso indiscriminado de las nuevas tecnologías, y a proceder con cautela en su aplicación como una panacea para la resolución de todos los problemas.

4. Propósitos y características del proyecto.

Como bien sabemos, en el proceso de enseñanza de lenguajes de programación, se pretende entrenar al alumno, para que pueda desarrollar un conocimiento creativo, que le permita modelar problemas del mundo real y trasladarlos al dominio del ordenador, para obtener soluciones de

una manera más rápida y eficiente. Esto constituye sin lugar a dudas, el objetivo primordial que debe cumplir nuestro proyecto, pero también se nos presenta la oportunidad de lograr otros beneficios que no están enfrentados con el anterior, y que de alguna manera, pueden contribuir al logro de esta y de otras metas.

Por lo anterior, se ha considerado, que el proyecto debe incorporar las siguientes características como cualidades principales de su diseño.

4.1. Asegurar un modelo eficiente de enseñanza-aprendizaje.

Se pretende llevar a la práctica el esquema representado en la figura 1.

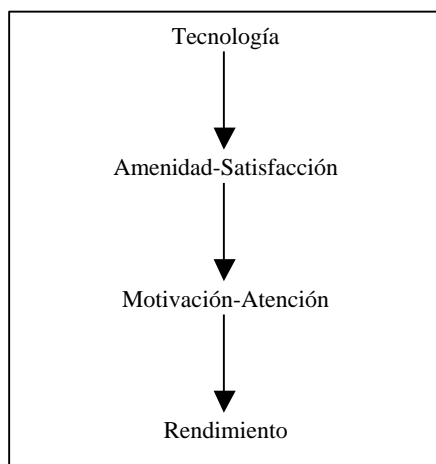


Figura 1. percepción del impacto de la tecnología en el proceso educativo.

Para lo cual se considera muy importante, basar la estrategia en un modelo que esté fundamentado en los siguientes puntos:

- Captar la atención del alumno, mediante una inducción que lo motive a aprender.
- Proponer actividades que varíen los estímulos y diversifiquen las formas de transmisión y adquisición del conocimiento.

- Emplear una comunicación efectiva, donde se incluye la sencillez en los términos, y el uso apropiado de tecnicismos.
- El uso efectivo de apoyos visuales, con un propósito definido, y apareciendo en el momento apropiado.
- La organización lógica (que incluye los objetivos, metodología y control del tiempo), incluyendo el manejo de preguntas, tanto las enfocadas a evaluar el desempeño, así como las que deberán contestarse a los alumnos.
- Situar al alumno, haciéndole saber siempre, qué actividad está realizando y las diversas opciones con las que cuenta.

4.2. Facilitar la labor administrativa del profesor.

Existen numerosas herramientas para la administración y evaluación del trabajo del estudiante a distancia [3,11]. El sistema proporcionará al profesor diversas utilidades enfocadas a llevar a cabo de una manera sencilla y organizada, el control de los aspectos académicos y administrativos del curso.

Esto le permitirá, tanto a él como a la administración de la universidad, así como de manera restringida a los mismos alumnos, conocer información estadística diversa, sobre el desempeño académico individual o colectivo del curso, además de facilitar los procesos administrativos de inscripciones, listas, pagos etc.

4.3. Crear un herramienta flexible e independiente del lenguaje de programación.

La adaptabilidad del sistema, para servir a la enseñanza de varios lenguajes de programación sin importar su paradigma es un reto importante en una asignatura como “*Programación lógica y funcional*” en la que, actualmente, se imparten dos lenguajes de programación (Prolog y Haskell) de dos paradigmas diferentes. Esto implica un diseño flexible, que permita cambiar la configuración de las herramientas utilizadas por el sistema como, por ejemplo, los bancos de ejercicios y pruebas, el manejo de interpretes, y sus interfaces con las diversas herramientas que controlarán la transmisión de la información a través de Internet.

4.4. Desarrollar interfaces adaptadas a las características individuales de cada alumno.

Todos los seres humanos, aunque poseemos cualidades más o menos homogéneas, contamos con ciertas cualidades o destrezas desarrolladas en mayor o menor medida, lo cual implica que las interfaces de usuario provistas en el entorno del sistema, deben acoplarse a dichas diferencias, de modo que cualquier persona sea capaz de utilizar la herramienta sin problemas. Atendiendo a lo anterior, se debe procurar que tanto el diseño de las páginas web y el entorno iconográfico que en ellas se maneje, así como el ambiente gráfico con que contará el alumno para desarrollar sus programas de práctica o de examen, deben adaptarse a sus propias habilidades o preferencias.

Ante todo, será importante ofrecer un entorno amigable y sencillo, que permita a los usuarios realizar las actividades que deseen sin contratiempos ni dificultades. El grupo Oviedo3 ha desarrollado diversas herramientas que permiten comprobar la usabilidad de sitios Web de forma remota [7]. La incorporación de dichas herramientas permitirá obtener datos sobre la interacción realizada por los usuarios y favorecerá la calidad del sistema.

4.5. Propiciar ambientes de trabajo en equipo.

Aunque la educación a distancia, se percibe como una herramienta para trabajar de manera aislada, las herramientas que nos proporciona para interactuar con otros usuarios, como el correo electrónico, *chat*, grupos de discusión, etc. ofrecen la posibilidad de realizar un sistema de creación colaborativa de información [10].

En el caso de lenguajes de programación, las peculiaridades de los mensajes de error de cada sistema y las múltiples posibilidades que aparecen al realizar programas, hacen necesario el desarrollo de sistemas colaborativos de asistencia al programador. No es de extrañar, la proliferación de los denominados sitios Wiki dedicados a diferentes lenguajes de programación, en los que los propios usuarios pueden añadir información de forma interactiva. En el proyecto se pretende incorporar un sistema similar de asistencia a los estudiantes en el que ellos mismos puedan añadir información de ayuda.

4.6. Proporcionar ayuda en línea

La oportunidad al prestar la ayuda, se transforma en un elemento fundamental para el aprovechamiento del tiempo y la asimilación de conocimiento. Se contará con un sistema de ayuda automática, que podrá accederse para comparar los problemas que se presenten al alumno, contra los que aparecerán un banco histórico, que almacenará los problemas o preguntas más frecuentemente solicitadas, además de contar con una bibliografía de consulta, para extender los conocimientos en algunos temas en los que se desee profundizar.

También es posible contar con asesorías sincrónicas, mediante videoconferencia o *chat*, que pueden programarse en horarios preestablecidos y asesoría asíncrona por medio del correo electrónico.

4.7. Implementación mediante servicios web.

La implementación de una herramienta flexible, que pueda ser ejecutada desde cualquier ordenador y además transportada a diferentes plataformas, exige el uso de herramientas estándar, soportadas por los servicios que proporciona Internet. En la actualidad, el desarrollo de aplicaciones sobre Internet tiende a basarse en los denominados servicios Web, que permiten descentralizar el control de determinados componentes de las aplicaciones facilitando una mayor versatilidad.

4.8. Internacionalización de la información

Hasta hace unos años, gran cantidad de software se producía pensando en satisfacer las necesidades locales o de un mercado reducido, y por lo tanto en un solo lenguaje. En la actualidad, el uso potencial por usuarios que manejan otros idiomas, y que comparten sus intereses educativos en materias como los lenguajes de programación, representa un aspecto muy importante, que motiva a afrontar el reto de hacer los programas más útiles, confeccionándolos en la lengua y cultura de cada colectivo potencial de usuarios.

4.9. Compatibilidad con los sistemas existentes

Como se ha mencionado, el sistema desarrollado debe ceñirse a su incorporación en un sistema existente. En la actualidad, se utiliza el sistema WebCT, pero es necesario tener en cuenta la posible utilización de otros estándares de enseñanza a distancia. Existen numerosas propuestas de estandarización de material de enseñanza a distancia [1, 13, 14] que deben tenerse en cuenta en la elaboración de material docente.

4.10. Evaluación automática del trabajo del alumno.

Sin duda alguna, la evaluación representa un punto clave del proceso de enseñanza-aprendizaje; para los maestros, este proceso de calificar un trabajo o bien un examen, en donde no existe una respuesta concreta es una tarea ardua cuya complejidad aumenta en relación con el número de alumnos a los que se desea evaluar.

De forma particular, los profesores de lenguajes de programación, tienen un gran reto tratando de asignar una nota cuyo valor refleje realmente la cantidad y calidad de los conocimientos adquiridos por el alumno durante el curso.

Llevar a cabo lo anterior de manera automática, es quizás, la parte más difícil de implementar. Teniendo en cuenta que la mayoría de las pruebas que se realizarán en este curso serán del tipo de construir un programa, el dominio de las respuestas que puede considerarse correctas resulta infinito. Una posibilidad teórica sería la utilización de sistemas de interpretación abstracta [15] que analicen ciertas componentes de los programas elaborados por los estudiantes, sin necesidad de ejecutarlos. Sin embargo, este tipo de análisis requiere un estudio de la semántica dinámica de los lenguajes implicados y en la actualidad sólo podría aplicarse a subconjuntos de dichos lenguajes. Este estudio semántico ha sido realizado de una forma modular en [16] y se considera una futura línea de investigación.

De una forma más pragmática, la evaluación de ejercicios de programación de forma automática se ha realizado en [18]. En el proyecto IDEFIX, se desarrollará un sistema basado en la especificación, para cada ejercicio de programación, de un conjunto de pruebas que

relacionan los datos de entrada con la respuesta a obtener. Algunas de estas pruebas serán visibles a los estudiantes, para que puedan validar ellos mismos sus programas. Otras pruebas serán ocultas, para evitar posibles trampas y permitir una evaluación automática. De la misma forma, se incorporará un sistema de chequeo automático de posibles copias o plagios.

5. Arquitectura del Sistema.

La arquitectura del sistema se presenta en la figura 2. Se mantienen los nombres en inglés de los diversos componentes para facilitar su reconocimiento en la implementación real. Los principales componentes son:

- **Problem Manager:** Forma el núcleo del sistema. Se comunica a través de Internet con los usuarios (alumnos, profesores y administradores) y gestiona la realización de los problemas de programación en los diferentes intérpretes y máquinas. Para ello, se crean diferentes procesos para la ejecución de cada problema utilizando un protocolo de comunicación con dichos procesos.
- **X Wrapper:** Encapsula la funcionalidad mínima de un intérprete genérico. Dicha funcionalidad viene dada por primitivas del tipo: *load* (carga un programa), *execute* (ejecuta un programa), *add* (añade un componente a un programa), *abort* (aborta la ejecución de un programa), etc. Se implementará un wrapper por cada uno de los lenguajes de programación que deseen incorporarse al sistema. En la actualidad se contempla la incorporación de los lenguajes *Haskell* y *Prolog*.
- **Machine (EXE):** El protocolo de comunicaciones con el *problem manager* también admite la ejecución de programas ejecutables distribuidos (componente *Machine*) que forman uno de los requisitos para impartir la asignatura de Sistemas Distribuidos.
- **Web based IDE:** Se encarga del interfaz de usuario y de la creación de un entorno de desarrollo amigable. El entorno se adaptará a las necesidades específicas de los usuarios.
- **Collaborative Programmer Assistant:** Se encarga de la creación de un sistema de

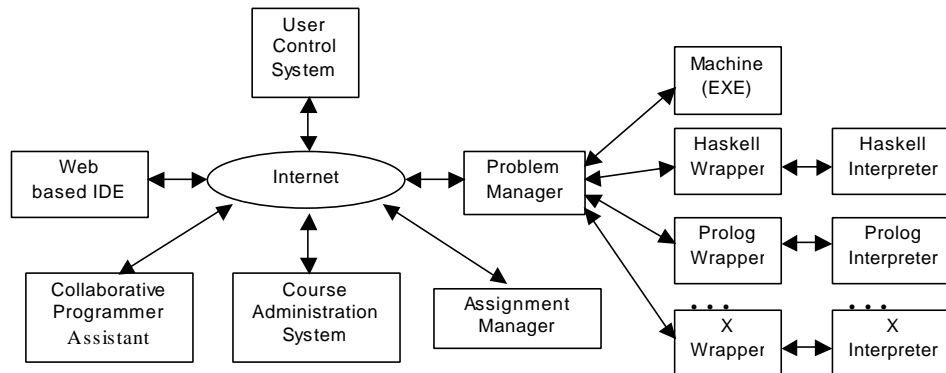


Figura 2. Arquitectura del sistema

ayuda a los programadores. Como se ha mencionado, se potenciará el desarrollo colaborativo por parte de los propios usuarios.

- **Course Administration System:** Este sistema realizará las tareas administrativas relacionadas con el curso. Estas tareas se comunican con el sistema de gestión académico general de la universidad. Por otro lado, este sistema llevará a cabo la monitorización de las diferentes peticiones realizadas por los estudiantes, facilitando la posterior evaluación. El sistema tendrá en cuenta, por ejemplo, el número de veces que un estudiante carga o ejecuta un programa, el número de programas erróneos y el tipo de errores, la participación de los estudiantes en los diversos foros, etc.
- **Assignment Manager:** Subsistema encargado de la gestión de las prácticas y problemas de programación. Aunque existen diversas herramientas para la publicación de ejercicios [8], en la actualidad se ha optado por la creación de un sistema propio basado en XML que permite la transformación automática de los enunciados a HTML y a LaTeX, permitiendo a la vez una visualización interactiva a través del Web y una impresión de calidad. El sistema admite el envío y presentación de enunciados de programación de forma temporal así como la recepción y evaluación de respuestas.
- **User Control System:** Sistema de control de usuarios. La principal novedad de este sistema es que no está empotrado en la aplicación, sino que se basa en la utilización

de un servicio Web similar al servicio Passport de Microsoft. De esta forma, se facilita la compatibilidad con la posterior adopción de sistemas integrales de seguridad por parte de la Universidad.

Debido a la heterogeneidad del equipo IDEFIX en el proyecto se ha adoptado un sistema descentralizado de control de versiones basado en SourceForge y se utiliza el idioma inglés para la especificación y codificación. Por otro lado, el software y las especificaciones desarrolladas mantienen una licencia de software libre.

6. Conclusiones

En los últimos años, los sistemas de enseñanza a través de Internet han surgido de forma espectacular. En el caso de la enseñanza de lenguajes de programación, aparecen unas necesidades específicas que suponen un reto para este tipo de sistemas.

En este artículo se han justificado las principales características que debe tener un sistema de realización de prácticas de programación a través de Internet y se ha presentado la arquitectura del sistema que en estos momentos se está desarrollando en el proyecto IDEFIX.

Un objetivo principal de la arquitectura presentada es su aplicación genérica, tanto a múltiples lenguajes y paradigmas de programación, como a diferentes asignaturas. De hecho, este tipo de herramientas, también podrían ser utilizadas por empresas de desarrollo de

software, ya que facilitan la creación de sistemas colaborativos y permiten evaluar el rendimiento de los programadores.

Entre las futuras líneas de investigación se considerará: la utilización de técnicas semánticas que permitan evaluar de forma automática el trabajo desarrollado sin necesidad de ejecutar los programas (con los consiguientes riesgos de seguridad), y apoyándose en herramientas de usabilidad, se realizarán investigaciones que conduzcan a la implementación de un entorno de desarrollo amigable.

Finalmente, aunque en la actualidad el proyecto está todavía en una fase preliminar, será necesario evaluar y comparar el rendimiento académico entre estudiantes presenciales y estudiantes a distancia con el fin de establecer las pautas a seguir para mejorar el sistema desarrollado.

Referencias.

- [1] Advanced Distributed Learning. *Sharable Content Object Reference Model*
<http://www.adlnet.org>
- [2] Arturo Azcorra Saloña, Carlos J. Bernardos, Óscar Gallego Gómez, e Ignacio Soto Campos, "Informe Sobre el Estado de la Teleeducación en España", Departamento de Tecnologías de las Comunicaciones, Universidad Carlos III de Madrid, Enero 2001
- [3] K. M. Dawson-Howe, "Automatic Submission of Programming Assignments", SIGCSE Bulletin 27(4) Dec. 95 pp. 51-53
- [4] Centro Informático y de Comunicaciones del Edificio de Ingenierías, "La Tecnología Educativa en el CICEI: una Perspectiva Histórica", septiembre de 2001.
<http://innova.cicei.com/historia/>
- [5] T. Clear, A. Haataja, J. Meyer, J. Suhonen, S. A. Varden, "Dimensions of Distance Learning for Computer Education", ITiCSE 2000 Working Group Reports, SIGCSE Bulletin, Vol. 33 (2). Junio 2001
- [6] Hassan Fazel, Vicente Manzano, Fco. Javier Pérez, "Variables Contextuales de las Aplicaciones Multimedia en la Universidad", Facultad de Psicología, Universidad de Sevilla, Revista Electrónica de Metodología Aplicada, 2000
- [7] B. M. González R., Jose E. Labra G., Juan M. Cueva L. "Web navigability testing with remote agents". Second ICSE Workshop on Web Engineering. Lecture Notes in Computer Science, vol. 2016, pp. 311-323, Springer-Verlag, 2001.
- [8] C. Gregorio Rodríguez, L. Llana Díaz, P. Palao Costanza, C. Pareja Flores, R. Martínez Unanue, J. A. Velázquez Iturbide, "EXercita: Automatic Web Publishing of Programming Exercises", 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, SIGCSE Bulletin, Vol. 33(3), pp. 161 - 164, Sep. 2001
- [9] Bruno Guardia Robles, "Asesores Inteligentes Para Apoyar el Proceso de Enseñanza de Lenguajes de Programación", Tesis para la Maestría en Ciencias Computacionales del I.T.E.S.M., diciembre 1997.
- [10] M. Guzdial, "Use of Collaborative Multimedia in Computer Science Classes", 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, SIGCSE Bulletin, Vol. 33(3), pp. 17 - 20, Sep. 2001
- [11] Dorota M. Huizinga, "Identifying topics for Instructional Improvement through On-line tracking of programming assignments", 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, SIGCSE Bulletin, Vol. 33(3), pp. 129 - 132, Sep. 2001
- [12] Página del Proyecto IDEFIX.
<http://www.di.uniovi.es/aplt/idefix>
- [13] IEEE Learning Technology Standards Committee
<http://ltsc.ieee.org>
- [14] IMS Global Learning Consortium
<http://www.imsproject.org>
- [15] N. D. Jones, F. Nielson, "Abstract Interpretation: a semantics-based tool for program analysis". Handbook of Logic in Computer Science, vol. 4 Semantic Modelling. Oxford University Press, 1995.
- [16] J. E. Labra, M. C. Luengo, J. M. Cueva, A. Cernuda, "A Language Prototyping Tool based on Semantic Building Blocks". Computer Aided Systems Theory, EUROCAST 2001, Lecture Notes in Computer Science, vol 2178, Springer-Verlag

- [17] R. Pérez Suárez, A. J. López Menéndez, "Aulanet, una Experiencia de Aula Virtual", Departamento de Economía Aplicada, Universidad de Oviedo, septiembre 2000.
- [18] R. Saikkonen, L. Malmi, A. korhonen, "*Fully Automatic Assessment of Programming Exercises*", 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, SIGCSE Bulletin, Vol. 33(3), pp. 133 - 136, Sep. 2001