

SISTEMA DE INTERACCIÓN REMOTA CON ROBOTS MÓVILES VÍA INTERNET

A. Khamis F. Rodríguez R. Barber M. A. Salichs

Departamento de Sistemas y Automática
Universidad Carlos III de Madrid
Avda, Universidad, 30 – 28911 Leganés, Madrid
{akhamis,urbano,rbarber,salichs}@ing.uc3m.es

Resumen: Este artículo presenta una arquitectura software para un laboratorio remoto en el campo de robótica móvil para facilitar la interacción remota con robots móviles utilizando varios dispositivos de interacción tales como las PDAs y los teléfonos móviles. En esta arquitectura se utiliza el paradigma del control supervisado en el desarrollo del sistema propuesto para disminuir el ancho de banda necesario y la sensibilidad del sistema al retraso temporal. Y se propone la utilización de los patrones de software para desarrollar interfaces hombre-robot flexibles y reutilizables. Las interfaces de usuario se han desarrollado utilizando el patrón *proxy* visual. Este patrón proporciona interfaces de usuario flexibles con relaciones de acoplamiento mínimas entre los subsistemas. La generación de la interfaz de usuario está completamente separada de los objetos de la capa de abstracción proporcionando así facilidad de reutilización y extensibilidad. Se ha integrado el laboratorio remoto desarrollado en un entorno educativo de robótica móvil que combina varias actividades para resolver los problemas del paradigma tradicional de instrucción y para permitir a los estudiantes tener un papel activo y creativo en el sistema. Se usa este entorno educativo como parte del proyecto IECAT [1], que pretende desarrollar una red de laboratorios remotos entre seis universidades en Europa y EE.UU. en el campo de los sistemas autónomos y teleoperados.

Palabras Clave: Robots Móviles, Laboratorios Remotos, Interfases Hombre-Robot, Teleducación, Interacción Remota.

I INTRODUCCIÓN

Las nuevas tendencias en robótica han ayudado a desarrollar muchos sistemas de robótica, que pretenden ampliar el uso de los robots en la vida diaria. La mayoría de estas nuevas tendencias se ha hecho posible gracias a la evolución de las tecnologías telemáticas.

Internet proporciona una infraestructura global de comunicación que habilita la implementación fácil de sistemas distribuidos. Aunque la red Internet mantiene un medio de comunicación barato y disponible, existen todavía muchos problemas por resolver antes de desarrollar aplicaciones verdaderamente fiables. Estos problemas incluyen el ancho de banda limitado y el retraso de la transmisión que varía arbitrariamente e influye en el rendimiento de los sistemas basados en Internet.

Recientemente, esfuerzos considerables de investigación en el campo de robótica móvil están dirigiéndose hacia el uso de Internet como medio de comunicación para facilitar la interacción remota con los robots móviles [2].

La experimentación remota tiene la ventaja de proporcionar la posibilidad de compartir un experimento o varios experimentos entre varios operadores localizados en lugares distintos. De esta manera, podrían compartirse fácilmente experimentos reales entre varios laboratorios y, por lo tanto, se podrían reducir los costes.

II INTERACCIÓN HOMBRE-ROBOT

La interacción hombre-robot (IHR) juega un papel importante en cualquier sistema de robótica teniendo en cuenta que todavía no existe un robot con capacidad totalmente autónoma. Incluso si esta meta de la autonomía completa se alcanzara, el papel humano y el nivel de interacción variarían pero el hombre seguirá siendo una parte del sistema. Por ejemplo, un robot personal interactuara regular y frecuente con un humano, mientras que un robot planetario de exploración recibiría comandos supervisados únicamente en casos ocasionales. La IHR puede definirse como el estudio de los seres humanos, los robots, y la forma en que uno influye en el otro [3].

La interacción remota es un tipo especial de la interacción hombre-robot, donde el hombre y el robot están separados por barreras físicas pero se comunican vía las tecnologías telemáticas. Se puede usar este tipo de interacción en muchas aplicaciones útiles como experimentación remota, teleoperación, telepercepción, teleprogramación, etc. Sin embargo, sin tener en cuenta la aplicación, todos los sistemas de interacción remotos con robots móviles suelen constar de los componentes básicos siguientes como se muestra en la figura 1 [4]:

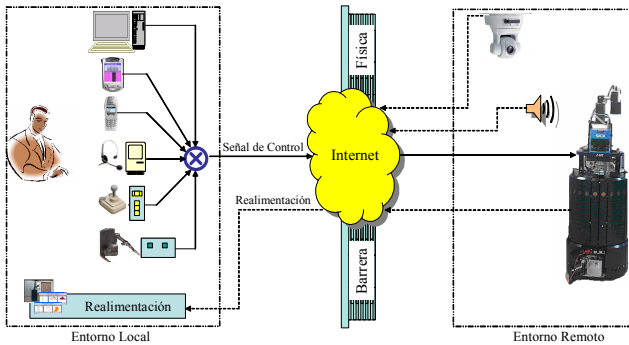


Fig. 1: Interacción Remota basada en Internet

- **Una interfaz de control:** Incorporando un dispositivo de interacción que el operador usa para mandar comandos al robot. Existen diferentes dispositivos de interacción con los que el usuario puede interactuar remotamente con el robot como el uso de ordenadores personales, el uso de dispositivos móviles como las PDAs y los teléfonos móviles, mediante comunicación hablada, el uso de joysticks y dispositivos de interfaz háptica.
- **Un robot móvil:** Que realiza las acciones ordenadas por el operador en el sitio remoto.
- **Un esquema de comunicación entre los dos sitios:** En la interacción remota basada en Internet, se usa Internet como medio de comunicación por medio del cual se transmiten las señales de control del operador al sistema del robot y luego se proporciona realimentación al operador desde el entorno del robot.
- **Interfaces de realimentación:** Se pueden usar varias herramientas para proporcionar realimentación al usuario. La realimentación más frecuente es la realimentación visual mediante la transmisión de vídeo. La transmisión de vídeo exige disponibilidad de un ancho de banda alto por eso se pueden utilizar los modelos gráficos y los modelos de realidad virtual como una alternativa para proporcionar la realimentación visual. Otro tipo es la realimentación audio que no exige ancho de banda elevado pero tiene la desventaja de ser sensible al retraso temporal de la comunicación. También las ayudas kinestéticas, como la realimentación háptica, pueden usarse para proporcionar una realimentación física adicionalmente a los otros tipos de realimentación. Utilizando la realimentación háptica, el usuario puede tener sensación táctil que expresa la secuencia de sus comandos directamente en la interfaz de control.

III LABORATORIOS REMOTOS

Los laboratorios remotos pueden ser una posible solución a los problemas que existen en los laboratorios experimentales universitarios. Un laboratorio remoto es un laboratorio basado en una red de comunicaciones, donde el usuario y los equipos del laboratorio están geográficamente separados y donde las tecnologías de las telecomunicaciones se usan para permitir a los usuarios acceder a los equipos del laboratorio. Este tipo de

laboratorios requiere una conexión permanente entre el usuario y el sitio Web durante el experimento y tiene la ventaja de no estar restringido a la asistencia sincronizada por instructores y estudiantes, es decir, tienen la potencia de proporcionar acceso constante siempre y cuando sea necesario.

Se pueden reunir varios laboratorios remotos formando un entorno de trabajo llamado laboratorio distribuido que se puede considerar como una red de laboratorios remotos. Un laboratorio distribuido es un entorno heterogéneo de resolución de problemas que permite a un grupo de investigadores, diseminados por todo el mundo, trabajar juntos en un conjunto común de proyectos mediante las tecnologías de información. Para ello se sirven, como en cualquier otro laboratorio, de las herramientas y técnicas específicas del dominio de investigación, pero los requisitos de infraestructura básica se comparten entre las distintas disciplinas. El proyecto IECAT en que participa la Universidad Carlos III de Madrid es un ejemplo de una red de laboratorios remotos en el campo de la mecatrónica aplicada a la robótica y las tecnologías aeroespaciales [1].

Especial interés tienen los experimentos basados en robótica móvil, puesto que representan sistemas reales complejos, que reúnen una gran variedad de conceptos. Para analizar la dificultad que significa elaborar un laboratorio remoto para robótica móvil se estudia la dificultad que conlleva la realización de distintos experimentos de forma remota.

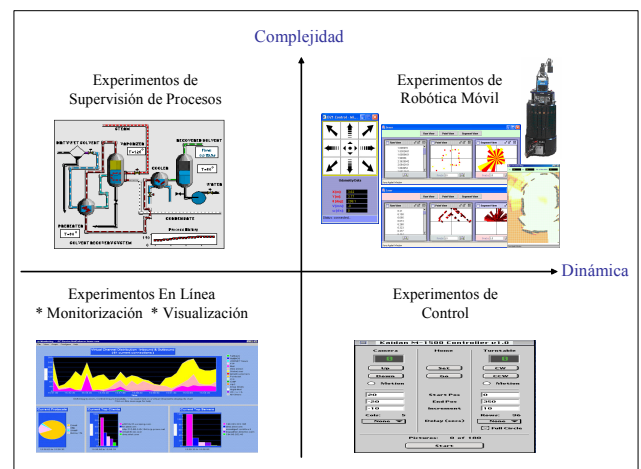


Fig. 2: Tipos de Experimentos remotos

Como se muestra en la figura 2, los experimentos remotos se pueden clasificar en los siguientes tipos:

- Experimentos en línea como monitorización o visualización. Son los más sencillos de implementar, puesto que son experimentos con escasa dinámica y baja complejidad.
- Experimentos de control de procesos, que aún careciendo de una dinámica importante, son mucho más complejos que los anteriores por el número de variables.

- Experimentos de control. Su dificultad viene dada por tener una alta dinámica, por lo que el retraso temporal que introduce Internet debe ser muy bien estudiado. El muestreo de las variables de salida para el posterior control de las variables de entrada ha de hacerse en tiempo real, para que el sistema no se desestabilice.
- Experimentos de robótica móvil que son los más complicados de desarrollar, por poseer una gran dinámica y tener una alta complejidad debido al alto número de variables a controlar. Los robots móviles van acompañados de gran número de sensores y actuadores sobre los que se puede actuar.

IV PATRONES DE SOFTWARE PARA INTERFACES HOMBRE-ROBOT

La interfaz es la parte, tanto hardware como software del sistema informático, que facilita al usuario el acceso a los recursos del sistema. Como se muestra en la figura 3, existen tres tipos de interfaces que tienen que ser desarrolladas para establecer la comunicación remota basada en Internet entre el ser humano y el robot.

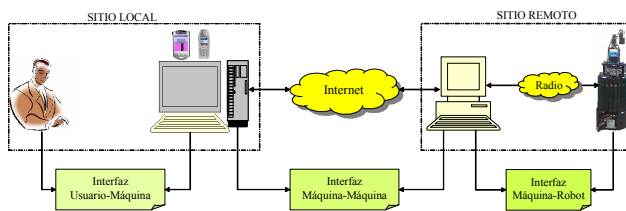


Fig. 3: Interfaces de un Sistema de Interacción Remota

Los tres tipos de interfaces se han desarrollado utilizando los patrones de software. Los patrones proponen una forma de reutilizar la experiencia de los desarrolladores. Para ello clasifican y describen formas de solucionar problemas que ocurren de forma frecuente en el desarrollo. Por tanto, están basados en la recopilación del conocimiento de los expertos en desarrollo de software. Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser utilizada más de un millón de veces, sin hacerlo ni siquiera dos veces de la misma forma [5]. A continuación, se explican los tres tipos de interfaces.

A. INTERFAZ USUARIO-MÁQUINA

La interfaz usuario-máquina o la interfaz gráfica de usuario es una parte fundamental en el proceso de desarrollo de los sistemas de interacción remota, por lo que desde el principio del desarrollo se debe tener en cuenta su diseño. La interfaz, como nexo de unión entre el ser humano y el sistema, se caracteriza por su apariencia (nivel de presentación), su capacidad de gestión del diálogo (nivel de control) y su funcionamiento (nivel de abstracción) [4]. La interfaz determinará en gran medida la percepción e impresión que el usuario poseerá del sistema puesto que el

usuario no suele estar interesado en la estructura y el funcionamiento interno de la interfaz. La tendencia actual a desarrollar interfaces intuitivas e independientes de la plataforma y del sistema operativo, está provocando que su diseño sea cada vez más complejo.

Se han desarrollado las interfaces de usuario utilizando el patrón proxy visual. El patrón “proxy visual” es una especialización de la arquitectura PAC (Presentación-Abstracción-Control) que define una estructura para sistemas de software interactivo como una jerarquía de agentes cooperantes como se ha mencionado anteriormente. Cada agente es responsable de un aspecto específico de la funcionalidad [6].

Al igual que la arquitectura PAC, la arquitectura de proxy visual consta, como se muestra en la figura 4, de los niveles de Presentación, Abstracción y Control.

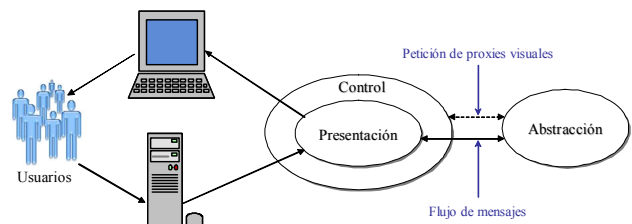


Fig. 4: El Patrón Proxy Visual

Este patrón permite la construcción de interfaces de usuario para sistemas orientados a objetos y garantizando la extensibilidad y la reutilización de los componentes del software implementados. Las tres capas que forman la arquitectura del proxy visual son:

- **Capa de Abstracción:** Es el grupo de objetos que directamente modela los conceptos abstractos en el dominio del problema. Esta capa es equivalente al modelo en la arquitectura MVC (Modelo-Vista-Controlador).
- **Capa de Presentación:** Son las clases que se encargan de dibujar la pantalla. Esas clases se pueden considerar como *proxies* visuales porque ellas realmente son *proxies* – representación – de los objetos de la capa de abstracción. En esta capa se gestionan las entradas y las salidas.
- **Capa de Control:** Pide los *proxies* visuales desde la capa de abstracción para construir la pantalla vista por el usuario. El objeto de control, entonces, está encargado de formar la interfaz de usuario.

B. INTERFAZ MÁQUINA-MÁQUINA

La interfaz máquina-máquina se refiere a los protocolos y los programas necesarios para intercambiar información a través de Internet entre dos máquinas ubicadas en lugares distintos. Se suele utilizar el modelo de programación distribuida cliente-servidor para llevar a cabo dicha comunicación. En este modelo la máquina del usuario funciona como un cliente que manda peticiones utilizando uno de los protocolos de Internet a otra máquina que

funciona como servidor. En los sistemas de interacción remota basados en Internet, se suele utilizar un applet de Java como cliente y un servlet como un servidor y en otros casos como servidor intermedio que reenvía las peticiones del cliente a otro servidor remoto como los servidores del robot. Un servlet es un programa escrito en Java que se ejecuta en el marco de un servicio de red (un servidor Web, por ejemplo), y que recibe y responde a las peticiones de uno o más clientes. Un servlet puede manejar múltiples peticiones concurrentes y puede sincronizarlas. Se usan peticiones/respuestas basadas en el protocolo HTTP para intercambiar información entre el cliente y el servidor.

C. INTERFAZ MÁQUINA-ROBOT

La interfaz máquina-robot es un caso especial de la interfaz máquina-máquina. Las dos interfaces se pueden desarrollar utilizando el modelo de programación distribuida cliente-servidor, pero la diferencia viene del hecho de que los dos programas del cliente y el servidor, que en este caso son el servlet y el servidor remoto del robot, están escritos en diferentes lenguajes de programación (Java y C++). Además, se usa la comunicación radio como medio de comunicación entre el cliente y el servidor, lo que provoca la necesidad de cambiar el protocolo de comunicación. En este caso, se puede usar CORBA para comunicar los servlet de Java con los servidores C++ del robot utilizando un protocolo de interoperabilidad llamado IIOP (Internet Inter-ORB Protocol), que permite a los clientes usar productos CORBA de cualquier desarrollador que se comuniquen con

objetos CORBA de cualquier otro desarrollador. Este modo de comunicación entre objetos distribuidos está basado en la implementación de un patrón llamado broker o intermediario, que facilita la comunicación entre objetos remotos en los siguientes casos:

- Los objetos deben acceder a servicios de otros objetos remotos pero sin preocuparse de su ubicación dentro de la red.
- Se añaden, cambian o modifican objetos en tiempo de ejecución.
- Los detalles de la implementación de los objetos deben de ocultarse a los clientes.

V ARQUITECTURA SOFTWARE

Una arquitectura software es una estructura de alto nivel de un sistema informático que comprende el conjunto de decisiones importantes sobre la organización del sistema [7]. Como se puede ver en la figura 4, la arquitectura propuesta está basada en el modelo cliente servidor de tres capas [4]. En esta arquitectura se usa el concepto de control supervisado por el cual el usuario tiene un papel de supervisión para mejorar el rendimiento del sistema y evitar los problemas de inestabilidad. Mediante el control supervisado, el usuario puede comunicarse con el sistema remoto a mayor nivel de abstracción usando comandos de alto nivel enviados al robot, el cual tiene un nivel mayor de autonomía para reducir la cantidad de datos que hay que transmitirse al sitio remoto.

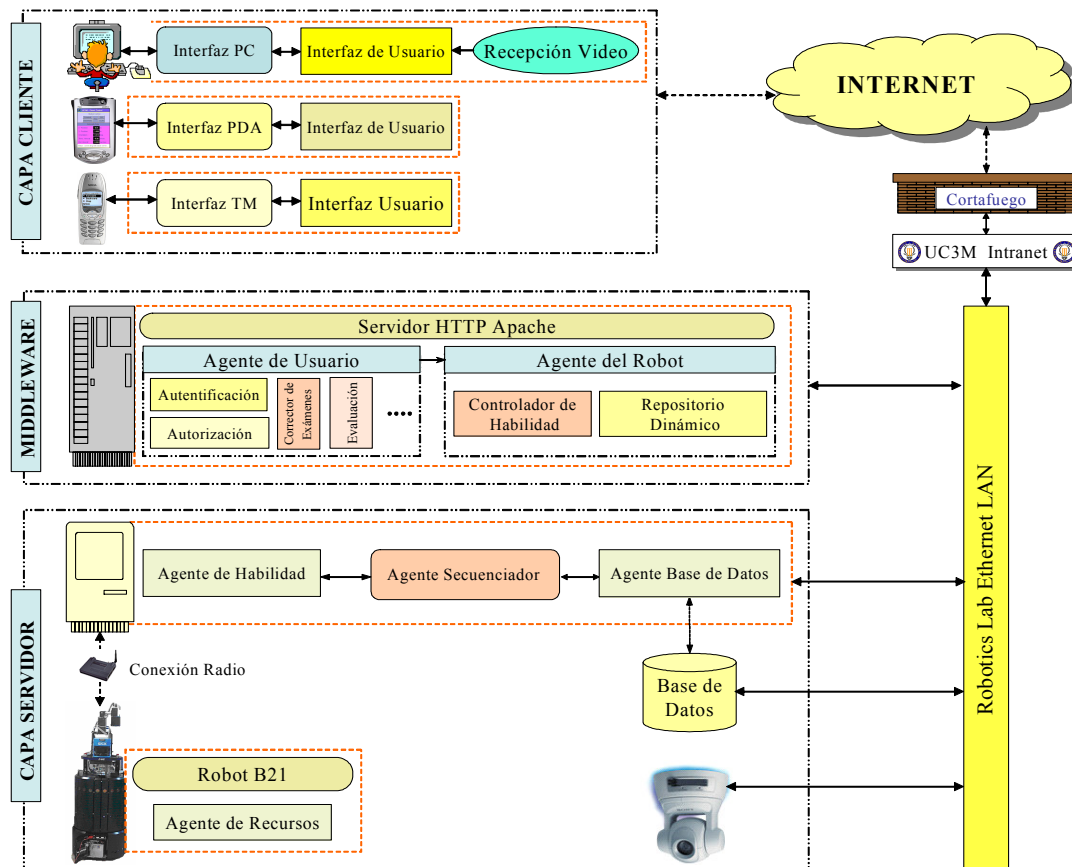


Fig. 4: Arquitectura del Sistema

El nivel de autonomía del robot se incrementa encapsulando todos los comportamientos del robot en varios tipos de habilidades. Limitar la interacción remota a comandos de alto nivel ayuda a reducir el ancho de banda necesario e incrementar la autonomía del robot puede ayudar a disminuir la sensibilidad de la tarea teleoperada al retraso temporal. A continuación, se explican brevemente las tres capas que forman esta arquitectura.

A. CAPA DE CLIENTE

Como muestra la figura 4, en la capa del cliente se adapta el diseño de la interfaz según las características del dispositivo de interacción. Por ejemplo, en el caso de dispositivos móviles como las asistentes digitales personales (*PDA- Personal Digital Assistant*), el mismo diseño puede ser usado con algunas modificaciones necesarias para tratar con los problemas específicos relacionados con las limitaciones de los dispositivos de mano (procesadores lentos, memoria limitada y pequeñas pantallas) y de redes inalámbricas (ancho de banda limitado, alta latencia, baja estabilidad de la conexión y baja disponibilidad) [4]. Se han desarrollado las interfaces de usuario utilizando la arquitectura de *proxy* visual explicada anteriormente. El objetivo del patrón proxy visual es separar completamente la generación de la interfaz de usuario de los objetos de la capa de abstracción para proveer facilidades de reutilización y extensibilidad.

B. CAPA DE MIDDLEWARE

La capa middleware se considera como una infraestructura lógica que proporciona los mecanismos básicos de direccionamiento y transporte entre la capa del cliente y la del servidor. En esta capa, hay un PC conectado internamente con una LAN Ethernet y externamente con la Intranet de la universidad, ejecutando un servidor Web apache, en el que se hospedan dos agentes como se muestra en la figura 4.

- **Agente de Usuario**

El agente de usuario ofrece varios servicios como el manejo de acceso de los usuarios al sistema y los servicios necesarios para el entorno de teleeducación. Este agente con la capa de cliente forman un modelo cliente-servidor de dos capas.

- **Agente del Robot**

El agente del robot trata con el robot real, durante su funcionamiento. Este agente provee la información requerida para procesar cierta habilidad seleccionada por el usuario o requerida en un paso de un experimento. Contiene dos grupos de servlets Java, los servlets de los controladores de las habilidades y los servlets del repositorio dinámico. Un servlet de controlador de habilidad recibe peticiones del usuario (activar/desactivar la habilidad), localiza el servidor remoto de la habilidad correspondiente en el agente de habilidad de la capa de servidor utilizando CORBA y manda las peticiones a dicho servidor para activar o desactivar la habilidad. Los servlets del repositorio dinámico facilitan la adquisición remota de

la información sensorial conectando el cliente con los servidores de los sensores en el agente de recursos de la capa de servidor. CORBA se usa como arquitectura de comunicación entre los objetos distribuidos de la capa de middleware (los servlets en Java) y los objetos de la capa de servidor (los servidores en C++).

C. CAPA DE SERVIDOR

La capa del servidor contiene cuatro agentes principales como se muestra en la figura 4.

- **Agente de Habilidad**

El agente de la habilidad representa el servidor de la habilidad, el cual según la arquitectura AD [8], podrá ser un servidor de una habilidad automática simple o de una habilidad deliberativa. Las habilidades simples pueden ser combinadas para formar habilidades complejas usando el agente secuenciador.

- **Agente Secuenciador**

Una de las tareas principales que se realiza dentro de un robot móvil, es la ejecución de tareas a partir de planes predefinidos. Un plan se define como el conjunto de habilidades (motoras o sensorimotoras) que deben ejecutarse en un momento dado a fin de conseguir un objetivo. Se puede definir el secuenciador como el motor de inferencia encargado de la correcta ejecución de un plan [9].

- **Agente de Base de Datos**

La secuencia de una habilidad compleja puede ser generada de dos formas, estas son almacenando la secuencia en un archivo de texto siguiendo las reglas gramaticales de un lenguaje libre de contexto o utilizando una herramienta gráfica para su construcción. Ambas formas necesitan de una base de datos para almacenar y gestionar los datos de una secuencia cuando se analiza. Para lograr todo lo anteriormente expuesto, se propone utilizar un agente manipulador de la base de datos encargado de:

- Gestionar y almacenar los datos y restricciones de cada habilidad simple y compleja.
- Permitir almacenar y verificar la validez de las nuevas secuencias.
- Responder a las peticiones realizadas por la clase *NetImpl* para cada secuencia que se este ejecutando en el robot [9].

- **Agente de Recursos**

El agente de recursos representa los servidores de los recursos del robot implementado en C++ como el servidor de la base (servidor del actuador, servidor de la odometría y servidor del sonar), el servidor del láser, etc. Se usa el robot b21 y el *framework* Mobility del robot que proporciona los módulos software básicos. Los servidores de los sensores o los sensores virtuales proporcionan a las habilidades automáticas y acciones refleja los datos leídos de los sensores físicos del robot [10]. Estos sensores informan sobre el entorno en el que el robot se mueve (Ej. sensores de ultrasonidos, sensor del láser, sensores de

También, como trabajo futuro, se pretenden desarrollar actividades de construcción locales como diseñar y construir nuevos prototipos o aplicar ingeniería inversa a sistemas existentes. Estas actividades se pueden usar para proporcionar la interacción cara-a-cara entre los estudiantes y los tutores y entre el estudiante y la máquina, y por consiguiente se aumenta la motivación de los estudiantes en la participación en el proceso educativo y se disminuye la sensación de aislamiento que algunos sienten al usar los sistemas de educación a distancia totalmente basados en Web. Ayudan también a aumentar la creatividad de los estudiantes y el trabajo en equipo. Para desarrollar estas actividades, se puede pedir a los estudiantes que construyan prototipos robóticos utilizando componentes hardware como Lego o Fischertechnik como componentes del hardware y los microcontroladores como Motorola

88HC11 o 16-bit Siemens 80C167 y utilizar varios sensores disponibles para los robots móviles.

El entorno educativo propuesto se usa, a partir del año académico 2001-2002, para modernizar una asignatura de doctorado de robots autónomos inteligentes. Los participantes en la asignatura (12 estudiantes) han evaluado el sistema mediante una encuesta en línea. Generalmente la impresión de los estudiantes ha sido positiva especialmente respecto al uso del laboratorio remoto. La mayoría de los estudiantes sentían que los experimentos en línea les ayudaron a lograr un entendiendo más profundo. La tabla 1 muestra la evaluación del sistema por parte de los estudiantes.

Tabla 1: Evaluación del Sistema

Pregunta	De acuerdo	Muy de acuerdo
El curso ha sido bien presentado	29%	71%
El sistema ha sido esencial par el curso	14%	86%
El sistema complementa el curso	43%	57%
EL sistema ha sido fiable	43%	57%
EL sistema ha sido fácil de usar	29%	71%
El laboratorio ha sido accesible	29%	71%

La apreciación de los participantes, que queda almacenada en el servidor Web, indica que los principales objetivos tanto del curso, como del laboratorio remoto han cumplido las expectativas.

VII CONCLUSION

Se ha desarrollado este sistema en el marco del proyecto IECAT [1] que pretende desarrollar una red de laboratorios remotos entre seis universidades en Europa y EE.UU. en el campo de los sistemas autónomos y teleoperados. Este laboratorio distribuido internacional va a permitir a los estudiantes llevar a cabo experimentos con equipamiento real y simulado perteneciente a las universidades participantes. Debido a la larga distancia que separa unas universidades de otras, los estudiantes tendrán a su disposición todo lo necesario para la teleoperación de robots y otros experimentos mediante laboratorios remotos, accesibles vía Internet. El sistema propuesto va a ser la aportación principal de la Universidad Carlos III de Madrid en el proyecto IECAT.

Agradecimientos

Los autores agradecen la financiación de los proyectos TAP 199-214 y DP12002-00188 ofrecida por el ministerio de ciencias y tecnología. Agradecemos también la financiación del proyecto IECAT (Innovative Educational Concepts for Autonomous and Teleoperated Systems) en el marco del programa de cooperación CE/EE.UU. en materia de enseñanza superior y formación profesional.

Referencias

- [1] IECAT (Innovative Educational Concepts for Autonomous and Teleoperated System) Web Page, 2004. <http://www.uc3m.es/isa/iecat.html>
- [2] *Archive of Online and Internet Robots*. 2004. Disponible en: <http://ford.ieor.berkeley.edu/ir/>
- [3] T. Fong, I. Nourbakhsh, y K. Dautenhahn, "A Survey of Socially Interactive Robots: Concepts, Design, and Applications", the Robotics Institute Carnegie Mellon University, Report CMU-RI-TR-02-29, 2002.
- [4] A. Khamis. Interacción Remota con Robots Móviles basada en Internet. Tesis Doctoral, Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, 2003.
- [5] Guía de Patrones de Diseño, Disponible en: <http://webs.teleprogramadores.com/patrones/>
- [6] A. Houlb, "Building User Interfaces for Object-oriented Systems", Parts 1:6, JavaWorld, July 1999, Disponible en: <http://www.javaworld.com/javaworld/jw-07-1999/jw-07-toolbox.html>
- [7] L. Bass, P. Clements, y R. Kazman. Software Architecture in Practice. 1st Edition, Addison Wesley Professional, 1998.
- [8] R. Barber. Desarrollo de una arquitectura para robots móviles autónomos. Aplicación a un sistema de navegación topológica. Tesis Doctoral, Departamento de Ingeniería Eléctrica, Electrónica y Automática (UCIIM), 2000.
- [9] D.M. Rivero, A. Khamis, F.J. Rodriguez, M.A. Salichs, "A Patterns-Oriented Framework for the Development of Automatic and Deliberative skills for mobile robots", The 11th International Conference on Advanced Robotics, ICAR03, University of Coimbra, Portugal, 2003.
- [10] M^a Jesús López. Sistema de Control para Robots Móviles Autónomos basado en Habilidades Reactivas. Tesis Doctoral, Departamento de Ingeniería Eléctrica, Electrónica y Automática, UC3M, 2002.