

Empleo de tecnologías de agentes para la gestión de tutorías en un campus universitario

Fabián Andrés Bustos
Dept. de Sistemas Informáticos y
Computación, D.S.I.C.
Universidad Politécnica de Valencia
46022 Valencia
fbustos@dsic.upv.es

Juan Sebastián López
Dept. de Sistemas Informáticos y
Computación, D.S.I.C.
Universidad Politécnica de Valencia
46022 Valencia
jlopez@dsic.upv.es

Vicente J. Julián
Dept. de Sistemas Informáticos y
Computación, D.S.I.C.
Universidad Politécnica de Valencia
46022 Valencia
vinglada@dsic.upv.es

Resumen

En la actualidad el empleo de la tecnología de agentes se ha convertido en una herramienta para encontrar solución a los problemas de cada día tratando de mejorar así la calidad de vida de las personas. El paradigma de los agentes nos permite crear sistemas que pueden ofrecer servicios complejos y sofisticados a diferentes tipos de usuarios. Utilizando plataformas como JADE-LEAP, podemos tener agentes ejecutándose en nuestro teléfono móvil o PDA, dándonos la capacidad de implementar sistemas flexibles, autónomos, abiertos, distribuidos y confiables. En este artículo describimos la arquitectura de un sistema multi-agente para ofrecer un Servicio de Gestión de Tutorías, que da la posibilidad a los usuarios (tanto alumnos como profesores), de administrar su horario de tutorías en la Universidad, de manera fácil, rápida y dinámica.

1. Introducción

Actualmente, con el desarrollo y evolución de la tecnología, vivimos en un mundo en donde es necesario poseer herramientas que nos permitan encontrar soluciones a los problemas cotidianos. La tecnología de agentes ha demostrado en los últimos años ser una herramienta confiable para encontrar soluciones a problemas en diferentes campos como medicina, turismo, recuperación de información, educación, etc. Podemos destacar algunos ejemplos en el área de turismo [1][2]. Ejemplos de otros tipos de aplicaciones pueden consultarse en [3]. Uno de los problemas que enfrenta actualmente la tecnología de agentes es aplicar los resultados teóricos obtenidos a lo largo

de los últimos años en problemas reales, aportando soluciones o servicios a usuarios.

En otro orden de cosas, en un futuro cercano se espera que las personas vean un incremento en la intrusión de dispositivos móviles/inteligentes en su vida diaria. Este artículo pretende testear el desarrollo de agentes en dispositivos móviles/inteligentes (tales como PDA o teléfonos móviles) desarrollando un problema distribuido real que requiere una gran cantidad de entidades o agentes. Así, este artículo describe una arquitectura multi-agente que permite ofrecer un servicio de gestión de tutorías en un campus Universitario.

Supongamos que un estudiante está en un campus universitario, y desea tener una tutoría con un profesor, en un primer lugar necesita saber donde se encuentra localizada la oficina del profesor para ir allí y pedir una cita. Después de que encuentre la oficina es posible que el horario de tutorías del profesor no esté disponible. Aun si el alumno puede acceder al horario de tutorías de dicho profesor, este puede no encontrarse en la oficina y no podrá establecer la cita con él.

Desde el punto de vista de un profesor universitario, sería de gran utilidad tener la información de sus tutorías disponible y actualizada. Es importante para el profesor gestionar su horario de tutorías, estableciendo restricciones y cambios cuando él lo considere necesario.

De esta forma, como podemos observar, existen muchos problemas relacionados a la manera tradicional de ofrecer un servicio de tutorías en un campus universitario. Todos estos aspectos han sido considerados para proporcionar una solución integrada empleando tecnologías de agentes, la cual será expuesta en las siguientes secciones. El resto del artículo está organizado de

la siguiente forma: la sección 2 provee una visión general del sistema propuesto, la sección 3 ofrece una visión general de la arquitectura del sistema y la relación entre los diferentes componentes, la sección 4 describe la implementación realizada. Finalmente, algunas conclusiones y trabajos futuros se muestran en la sección 5.

2. Descripción del sistema

Esta sección muestra la aplicación desarrollada en la cual se ha evaluado la aplicabilidad de la tecnología de agentes integrando diferentes dispositivos. En la versión actualmente implementada de este sistema, un estudiante puede consultar los horarios de tutorías de un profesor determinado de acuerdo con sus preferencias usando un teléfono móvil o una PDA. Una vez que ha seleccionado un horario, el estudiante puede establecer una reserva de un periodo de tiempo específico para su tutoría. Este proceso es establecido entre un agente interfaz estudiante y un agente gestor de horarios de tutorías. Además, el sistema implementado incluye más servicios y agentes que serán explicados en la sección 3.

2.1. Arquitectura del sistema

Para elaborar un diseño acertado del sistema, debemos identificar todos los componentes involucrados en la gestión de tutorías en un campus universitario. En primer lugar, el servicio está dirigido a dos tipos de usuarios, que son, profesores y estudiantes. Cada uno de ellos tiene diferentes necesidades y diferentes características. Debido a esto, se decidió incluir en el sistema dos agentes diferentes para la gestión fundamentalmente de interfaz con dichos usuarios: *Teacher Agent* y *Student Agent*.

Por otro lado, es importante poder tener acceso a la información en cualquier momento y lugar dentro del campus universitario. Por esta razón, tanto el *Teacher Agent* como el *Student Agent* fueron diseñados para ser usados sobre dispositivos móviles como un teléfono móvil o una PDA. Para un profesor es importante tener la capacidad de administrar la información acerca de sus tutorías de forma ordenada y eficiente. Además, la gestión del horario de un profesor debe tener continuidad temporal para poder

ofrecer sus servicios de forma ininterrumpida. Debido a esto, se planteó introducir otra entidad que administrase toda la información sobre los horarios de tutorías de un profesor específico, el *Tutorship Manager Agent*. Este agente debe mantener la información disponible todo el tiempo. Pensando en una arquitectura distribuida y eficiente, cada *Teacher Agent* tendrá un *Tutorship Manager Agent* asociado para manejar su horario de tutorías.

Considerando que los agentes deben ser capaces de ponerse en contacto de alguna forma entre ellos, necesitamos una entidad que ofrezca un conjunto de servicios que faciliten la comunicación a otros agentes usando algún conocimiento sobre los requerimientos y las capacidades de esos agentes [4]. Dicha funcionalidad la aportará la figura de un *Broker Agent*. El broker, además de las facilidades de comunicación, se encarga de realizar la búsqueda, procesar la información obtenida y enviarla directamente al estudiante interesado.

Resumiendo, el sistema está constituido básicamente por cuatro tipos de agentes: el *Broker Agent*, el *Teacher Agent*, el *Tutorship Manager Agent* y el *Student Agent*. Las funcionalidades principales de estos agentes son: (I) El *Broker Agent* contiene información actualizada de los *Tutorship Manager Agents* registrados con él. Este se encarga de establecer la comunicación entre el *Student Agent* y el *Teacher Agent*. (II) El *Tutorship Manager Agent* maneja toda la información acerca del horario de tutorías de un profesor específico. Será exclusivo a un profesor, y provee los siguientes servicios a los profesores: consultar, modificar y establecer restricciones en su horario de tutorías, y consultar las tutorías asignadas para un día en particular. (III) El *Teacher Agent* permite a los profesores acceder y utilizar los diferentes servicios ofrecidos por el *Tutorship Manager* mediante una GUI ejecutándose en su dispositivo móvil. (IV) El *Student Agent* será exclusivo para cada estudiante interesado, y provee los siguientes servicios: consulta de horario de tutorías por materia o nombre de profesor, reserva de tutorías y consulta de tutorías solicitadas para un día en particular. La comunicación entre el estudiante y su agente se llevará a cabo mediante una GUI asociada que el estudiante ejecuta en su dispositivo móvil.

Debido a que los agentes necesitan comunicarse entre ellos, fue indispensable

establecer un vocabulario conceptual común como una representación de la información para establecer y controlar tareas. Toda esta información queda representada dentro de una ontología específica. La ontología implementada proporciona una descripción detallada de los profesores, tutorías, estudiantes, horario de profesores, etc. Adicionalmente, la ontología desarrollada contiene acciones y predicados que ayudan a controlar y establecer tareas dentro del sistema.

La Figura 1 muestra la arquitectura del sistema multi-agente implementado ilustrando la comunicación entre los diferentes agentes comentados anteriormente. Básicamente, si el usuario (profesor o estudiante) se encuentra ubicado en cualquier punto dentro del campus, usando la red *wireless* puede utilizar el sistema de gestión de tutorías, enviando preguntas o acciones al *Broker Agent*. El *Broker Agent* establecerá una interacción con el *Tutorship Manager Agent* correspondiente intentando encontrar la información deseada del profesor. Posteriormente, el *Broker Agent* enviará la respuesta apropiada al agente profesor o estudiante interesado según sea el caso.

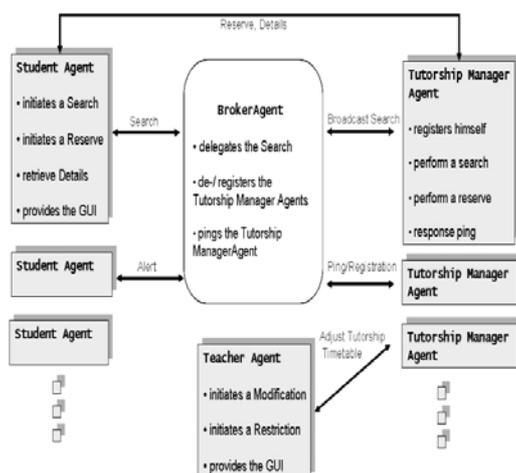


Figura 1. Arquitectura del Sistema multi-agente

Después de que el usuario obtiene una respuesta de su búsqueda, este establecerá una comunicación directa con el *Tutorship Manager Agent* correspondiente.

Los usuarios pueden establecer la conexión requerida usando el acceso *wireless* que se encuentra disponible en el campus universitario. La selección de este tipo de conexión se hizo debido a que esta cubre la mayor parte del campus y permite una velocidad de conexión apropiada. Obviamente esta opción es posible únicamente en dispositivos móviles como una PDA con acceso mediante *wireless* integrado. En el caso de teléfonos móviles, la conexión puede ser establecida mediante GPRS. Este proceso se ilustra en la Figura 2.

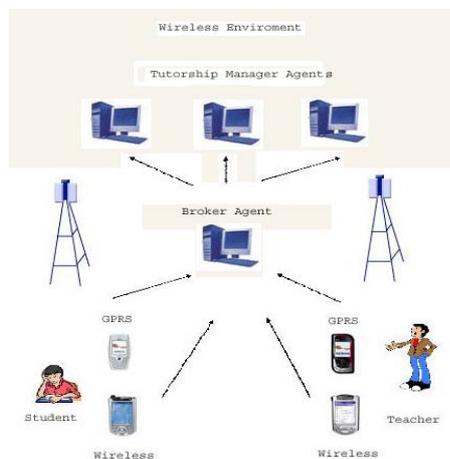


Figura 2. Esquema sistema ejecutándose a través de conexiones diferentes

2.2. Funcionalidad del sistema

Esta sección describe brevemente los servicios ofrecidos por los diferentes agentes dentro del sistema.

Consulta de horario

Este servicio es ofrecido por el *Tutorship Manager Agent* y puede ser invocado por el *Teacher Agent* y el *Student Agent*. Para poder ser utilizado, el *Teacher* o *Student Agent* debe enviar un mensaje QUERY (de acuerdo con las especificaciones de FIPA-ACL [5]) solicitando el horario requerido. El resultado de este servicio

será la información del horario de tutorías de un profesor específico.

Para consultar el horario de un profesor concreto, el estudiante puede realizar una búsqueda por nombre de profesor o asignatura. Esta búsqueda se realiza introduciendo una parte del nombre de una asignatura o del nombre de un profesor. El *Student Agent* enviará un mensaje al *Broker Agent*. El *Student Agent* recibirá un mensaje con una lista de los elementos que cumplen con sus parámetros de búsqueda. Al elegir uno de los elementos de la lista, el *Student Agent* intentará obtener el horario de un profesor, enviando un mensaje al *Tutorship Manager Agent* correspondiente. El *Tutorship Manager Agent* responderá con un mensaje de tipo INFORM, conteniendo la información del horario de tutorías al *Student Agent*.

Reserva

Este servicio es ofrecido por el *Tutorship Manager Agent* y puede ser invocado por el *Student Agent*. Para poder usarlo, el *Student Agent* debe enviar un mensaje tipo PROPOSE al *Teacher Manager Agent* con la propuesta de reserva de horario que quiere realizar el estudiante. El resultado de este servicio será una reserva de tutoría exitosa o un mensaje de error.

De forma más detallada, el *Student Agent* envía un mensaje que contiene información concerniente a la petición de una tutoría como hora y fecha, duración, etc. El *Tutorship Manager Agent* entonces verificará esta información, y si la reserva de la tutoría es posible enviará un mensaje al *Student Agent*. De lo contrario, el *Tutorship Manager Agent* intentará encontrar una hora o fecha alternativa en donde pueda cumplir con la petición del estudiante. Si es posible, el *Teacher Manager Agent* enviará una respuesta con la nueva tutoría propuesta. El *Student Agent* puede entonces aceptar o negar esta propuesta. Si el proceso de reserva es exitoso el *Tutorship Manager Agent* guardará todos los datos de la tutoría (nombre de estudiante, fecha, hora, asignatura, etc.) dentro de su base de conocimiento.

Modificación de horario de tutorías

Este servicio lo ofrece el *Tutorship Manager Agent*. Puede ser invocado únicamente por el *Teacher Agent*. Para poder ser utilizado, el *Teacher Agent* debe enviar un mensaje tipo REQUEST, con la información de las modificaciones que se desean establecer sobre el horario de tutorías existente. El resultado de este servicio será una actualización del horario de tutorías.

Cuando un profesor modifica su horario de tutorías, y tiene una o más tutorías asignadas en este horario, el *Tutorship Manager Agent* debe informar a los *Student Agents* correspondientes acerca de la cancelación de la tutoría. Además, el *Tutorship Manager Agent* espera por una notificación de que el mensaje ha sido recibido correctamente. Si el *Student Agent* no se encuentra disponible en el momento de la modificación, el *Tutorship Manager Agent* enviará periódicamente un mensaje de alerta hasta que el *Student Agent* sea informado.

Establecer restricciones al horario de tutorías

Este servicio lo ofrece el *Tutorship Manager Agent*. Este puede ser invocado únicamente por el *Teacher Agent*. Para poder ser utilizado, el *Teacher Agent* debe enviar un mensaje tipo REQUEST. Este mensaje debe indicar las características de la nueva restricción impuesta por el profesor. El resultado de este servicio será un horario de tutorías actualizado.

Este servicio permite al profesor tener la oportunidad de establecer restricciones en una hora o fecha específica de su horario. Un motivo de restricción podría ser un viaje, otro compromiso, etc. El sistema será capaz de readaptar el horario de este profesor a la nueva situación.

De igual forma que en el servicio de modificación de horario, cuando una tutoría que ha sido asignada se ve comprometida por una nueva restricción, se notificará al *Student Agent*.

Reserva on-line

Este servicio lo ofrece el *Tutorship Manager Agent* y puede ser invocado por el *Student Agent*. Para poder utilizar este servicio el *Student Agent*

debe enviar un mensaje tipo REQUEST, con información acerca de la reserva a solicitar. El resultado de este servicio será una reserva de tutoría exitosa o un mensaje de error.

El servicio de reserva on-line se ofrece por medio de una conexión Bluetooth desde la oficina del profesor. La principal ventaja de este servicio es ofrecer la posibilidad de reservar si el estudiante va a la oficina del profesor y el profesor no se encuentra disponible. Actualmente, casi todos los dispositivos móviles soportan Bluetooth. El sistema implementado da al estudiante otra posibilidad para hacer uso del servicio de reserva. Esto se hace a través de una conexión Bluetooth entre su PDA o teléfono móvil y un computador fijo. Buscando mantener un diseño óptimo del sistema, en el computador de cada profesor, en donde estará siempre ejecutándose el *Tutorship Manager Agent* han sido instalados nodos Bluetooth. Cuando un estudiante esté frente a la oficina de un profesor, el puede acceder a este servicio.

Otros servicios

Además de los servicios expuestos anteriormente, el sistema multi-agente cuenta con dos servicios básicos ofrecidos por el *Broker Agent*. Estos son el servicio de registro y de ping. Con el primero se garantiza el registro de los diferentes *Tutorship Manager Agent* dentro del sistema multi-agente y, el segundo verifica que los *Tutorship Manager Agent* previamente registrados continúan estando activos dentro de la plataforma.

3. Implementación

3.1. Plataformas de programación

Se ha implementado el Sistema de Gestión de Tutorías usando la plataforma de agentes JADE [6] que esta totalmente escrita en Java. JADE es un *middleware* que facilita el desarrollo de sistemas multi-agente. Este provee las funcionalidades típicas impuestas por FIPA como son el DF (Directory Facilitator), el AMS (Agent Management System), el canal de comunicación y una capa de protocolos de transporte de mensaje.

Los agentes ejecutándose en los dispositivos móviles fueron implementados usando el LEAP (*Lightweight Extensible Agent Platform*) *add-on* [7]. Este, combinado con JADE, proporciona un entorno de ejecución modificado llamado JADE-LEAP ("JADE powered by LEAP") que hace posible ejecutar agentes FIPA en dispositivos de pocas prestaciones ejecutando Java.

Las interfaces gráficas fueron desarrolladas usando la especificación J2ME [8] (Java 2 Micro Edition) que combina el perfil MIDP (Mobile Information Device Profile), con la configuración CLDC (Connected Limited Device Configuration). Ambos en conjunto, proveen una serie de funcionalidades requeridas por las aplicaciones Java para ser ejecutadas en dispositivos móviles, mediante un entorno de ejecución estandarizado Java y un completo grupo de APIs Java.

Para administrar y controlar la conexión Bluetooth, se ha usado el API de Java especificado para la tecnología inalámbrica Bluetooth (JSR 82) [9], un estándar para desarrollar aplicaciones de Bluetooth usando el lenguaje de programación Java.

3.2. Hardware

Como se comentó anteriormente, el sistema fue desarrollado para permitir la integración de dispositivos móviles. Por esta razón, se ha usado la especificación J2ME. Concretamente, se ha empleado el Wireless Toolkit 2.2 de dicha especificación. Esta versión soporta funcionalidades como el Wireless Messaging API, Mobile Media API, la especificación J2ME para Web Services y el API de Java para Bluetooth, siendo esta última especialmente útil dentro de nuestra aplicación y trabajo futuro. Este Wireless Toolkit provee un emulador que se ha usado para simular el teléfono móvil y la ejecución de los *Teacher Agents* y *Student Agents* dentro de este.

Se ha usado la iPAQ 5450 de Hewlett Packard que dispone de conexión Bluetooth y 802.11b integrada para ejecutar los *Teacher Agents* y *Student Agents* sobre una PDA. Esta Pocket PC esta basada en un procesador Intel XScale a 400MHz y cuenta con 64MB y el sistema operativo Microsoft Windows CE. Además, como máquina virtual de Java para nuestra Pocket PC hemos usado CrEme [10].

En los ordenadores donde el *Tutorship Manager Agent* esta ejecutándose, se han adherido adaptadores USB para Bluetooth de ANYCOM para permitirles comunicarse con otros dispositivos Bluetooth. Estos adaptadores Bluetooth poseen un rango de conexión (hasta 100 metros) apropiado para nuestro sistema.

4. Resultados

Es esta sección presentamos un escenario hipotético que ilustra la interacción entre el *Student Agent* y el *Tutorship Manager Agent* ejecutándose en un teléfono móvil. Supongamos que un estudiante está en el campus y necesita aclarar algunos aspectos en una asignatura concreta llamada “Agentes JADE”.

En primer lugar, el estudiante debe tener instalado el *midlet* en su teléfono móvil para ejecutar la aplicación de Gestión de Tutorías, como se muestra en la Figura 3.



Figura 3. Midlet de la aplicación gestión de tutorías

Al ejecutar la aplicación, el estudiante introduce su nombre de usuario y su contraseña usando la interfaz del teléfono móvil, pudiendo así tener acceso a los servicios ofrecidos por la aplicación (Figura 4).



Figura 4. Registro dentro del sistema

Después de que el estudiante se ha identificado, verá en la pantalla de su móvil un “Menu” con las diferentes opciones disponibles. Aquí, dicho estudiante puede realizar una búsqueda por asignatura, profesor o puede revisar sus tutorías previamente reservadas (Figura 5).



Figura 5. Pantalla de menú conteniendo los diferentes servicios ofrecidos por el sistema.

Como se mencionó anteriormente, el estudiante desea realizar una cita con un profesor específico, y conoce el nombre de la asignatura. Al seleccionar la opción del Menu de “Consulta por asignatura”, el estudiante introduce una parte del nombre de la asignatura, y de esta forma obtiene una lista con los resultados que corresponden con su parámetro de búsqueda. Cada resultado está constituido por el nombre de asignatura y el profesor que la imparte, como se muestra en la Figura 6.

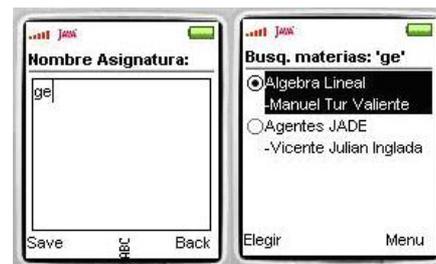


Figura 6. Búsqueda por asignatura y resultados obtenidos

Posteriormente, el estudiante puede elegir de esta lista el profesor del que desea obtener su horario de tutorías y, tal vez reservar una tutoría (Figura 7).



Figura 7. Horario de tutoría de un profesor

Ahora, el estudiante puede seleccionar el horario de tutorías que satisface sus necesidades. Finalmente, dicho estudiante decide que la mejor fecha para tener la tutoría es el 25 de Enero a las 10:30. En este caso la tutoría es asignada con éxito a este estudiante (Figura 8).

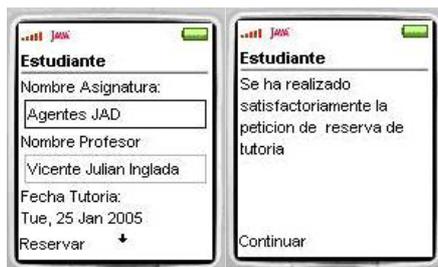


Figura 8. Proceso de reserva.

5. Conclusiones

En este artículo hemos presentado el diseño e implementación de un Sistema de Gestión de Tutorías para dispositivos móviles tales como PDA y teléfonos móviles.

El sistema implementado no es un sistema excesivamente complejo, pero es un sistema completamente implementado y testeado, cuyo principal objetivo es ser empleado como una prueba de la aplicabilidad de la tecnología de agentes integrando diferentes dispositivos como PDA o teléfonos móviles. Además, el sistema se puede extender con un proceso de negociación sofisticado entre el *Tutorship Manager Agent* y el *Student Agent*.

Esta aplicación es el punto inicial para desarrollar aplicaciones similares dentro del ambiente universitario, como reservar diferentes

elementos dentro de la universidad: equipo, libros, salas, instalaciones deportivas, etc.

Como trabajos futuros se desarrollará un *benchmark* para evaluar el sistema en términos de escalabilidad. Esto permitirá con posterioridad analizar y estudiar el rendimiento de la tecnología de agentes en sistemas reales de mayor escala, en donde un gran número de agentes se interconectan usando diferentes dispositivos.

Referencias

- [1] Corchado J. M., Pavón J., Corchado E. y Castillo L. F. (2004) Development of CBR-BDI Agents: A Tourist Guide Application. 7th European Conference on Case-based Reasoning 2004. Universidad Compluense de Madrid, Madrid, 30 Septiembre - 2 Octubre.
- [2] Bätzold, M.; Navarro, M.; Julián, V.; Botti, V. (2003): Desarrollo de servicios turísticos a usuarios. In Proceedings of the Conference of the Spanish Association for Artificial Intelligence (CAEPIA), San Sebastian, Spain, November 2003.
- [3] Ana Mas (2005): Agentes Software y sistemas multi-agente: conceptos, arquitecturas y aplicaciones. Pearson Educación, S.A. 2005.
- [4] FIPA Brokering Interaction Protocol Specification.
<http://www.fipa.org/specs/fipa00033/XC00033F.html>
- [5] Foundation for Intelligent Physical Agents, <http://www.fipa.org>.
- [6] Java Agent DEvelopment Framework. <http://sharon.cselt.it/projects/jade/>
- [7] Light Extensible Agent Platform. <http://leap.crm-paris.com/>
- [8] Java 2 Micro Edition <http://java.sun.com/j2me/>
- [9] <http://jcp.org/en/jsr/detail?id=82> Java APIs for Bluetooth Wireless Technology (JSR-82)
- [10] CrEme, the Java Virtual Machine for Windows CE type devices, <http://www.nsicom.com>