

Fridolin Wild, Steinn E. Sigurdarson

Vienna University of Economics and Business Administration (Austria)

<fridolin.wild@wu-wien.ac.at>,
<sesigurd@wu-wien.ac.at>

Redes de alimentadores distribuidos para el aprendizaje

Traducción: Agustín Palomar Ollid (Grupo de Trabajo de Lengua e Informática de ATI)

1. Aprendiendo con bitácoras

El conglomerado de todas las bitácoras (*blogs*) disponibles en línea, la llamada "blogosfera", se ha certificado que muestra una evolución explosiva al menos desde 2001, donde puede identificarse un aumento eruptivo no solo según las métricas de escala sino también con respecto a las cada vez más profundas estructuras de comunidad y a los grados de conectividad superiores [1][4]. Por ejemplo en diciembre de 2007, Technorati, el sitio para índice y búsqueda de bitácoras, está indexando más de 112 millones de bitácoras [2]. El uso de bitácoras es un fenómeno popular que se incrementa, a pesar de que hay meta-estudios que revelan que entre la mitad y los dos tercios de todos los *blogs* se abandonan apenas en los dos meses posteriores a su creación [3].

Una de las razones de que las bitácoras se hayan vuelto tan atractivas es su sencillez de uso, eliminando las barreras de los conocimientos técnicos que son precisos para publicar uno mismo en la web [5]. Hay una plétora de herramientas para publicar en web que permiten al usuario elegir entre una gran variedad de servicios de alojamiento no-comerciales (a menudo disponibles sin cargo). Más aún, los usuarios pueden crear sus propias aplicaciones web eligiendo del rico catálogo de productos de código abierto y cerrado. El Centro de e-Aprendizaje Learning Light, por ejemplo, listaba ya en el pasado 2006 más de 56 productos y servicios en línea diferentes en un directorio de proveedores de herramientas para *blogs* [6]. La publicación de contenido rico con *blogs* no requiere de ningún conocimiento técnico profundo como las habilidades de codificación necesarias para crear páginas con un editor html, o habilidades como las necesarias para crear un sistema de gestión de contenidos hecho y derecho.

A diferencia de las aulas virtuales, los *wikis*, o los foros, las bitácoras ofrecen intrínsecamente la opción de construir redes abiertas a la colaboración, sin necesidad de establecer previamente una comunidad dedicada y comprometida con la comunicación. Un único *blog* puede ser considerado como una publicación individual. La blogosfera como un todo es, sin embargo, participativa por naturaleza [20].

No sorprende entonces que las bitácoras se

Resumen: estudios recientes indican que las bitácoras (*blogs*) son la aplicación de usuario más innovadora de ésta década. Hasta ahora, la blogosfera en su forma actual padece varios problemas. Las ideas poco claras de la audiencia, la desconexión, la fragmentación, y la falta de coherencia conversacional pueden tener sus raíces no solo en factores sociológicos sino en deficiencias tecnológicas de la infraestructura actual. Estos problemas obstruyen el desarrollo efectivo de las bitácoras en actividades de aprendizaje colaborativo. Dentro de esta contribución, se propone una especificación de interfaz para la distribución centrada en el usuario de las actividades de agregación de alimentadores (*feeds*) que es tanto un prerrequisito como una infraestructura básica para la colaboración basada en bitácoras. Al presentar una visión general sobre el estado del arte en los estándares de alimentación e interacción, se mostrará una clara falta de apoyo para la gestión de red activa. Los requerimientos de diseño de una solución que llene este hueco serán bosquejados y complementados con una descripción paso a paso del proceso de comunicación de la especificación "FeedBack" (retroalimentación) propuesta. Los resultados preliminares de una prueba con una implementación de referencia para WordPress, proporcionan una demostración del concepto.

Palabras clave: alimentadores, aprendizaje potenciado por la tecnología, ICT, TEL.

Autores

Fridolin Wild es Master of Arts en Ciencias de la Información y Política. Actualmente, está investigando con ProLearn, la Red de Excelencia (NoE) de la Unión Europea (UE) para el aprendizaje profesional potenciado por la tecnología, y adicionalmente dentro del proyecto iCamp con fondos IST de la UE, donde es el gerente técnico y dirige un paquete de trabajo sobre la interoperabilidad de las herramientas de software social para el aprendizaje. Fridolin es el tesorero de la Asociación Europea de Aprendizaje Potenciado por la Tecnología (EATEL). Trabaja como científico en el Instituto de Sistemas de Información de la Universidad de Economía y Administración de Empresas de Viena.

Steinn E. Sigurdarson es investigador del aprendizaje potenciado por la tecnología en la Universidad de Economía y Administración de Empresas de Viena, trabaja en interoperabilidad y herramientas de software social para el aprendizaje dentro del proyecto iCamp. Steinn tiene experiencia en proyectos de integración de datos en el sector empresarial con un foco en desarrollos de software libre. Desde 2006 ha estado concentrado en el desarrollo de software social, entre otros cómo líder de fase en el desarrollo de software del proyecto Covcell con fondos Minerva.

conviertan en vehículos de gestión del conocimiento para constituir a menudo una parte integral de los procesos de enseñanza y aprendizaje. Los *blogs* pueden utilizarse para organizar conferencias, seminarios, y debates tanto entre profesores como entre estudiantes. Herring et al. (2004) hallaron en su estudio sobre los tipos de *blog* que el 57.5% de los autores de *blog* investigados en una muestra aleatoria de 203 *blogs* eran estudiantes de nivel secundario y terciario [7]. Sin embargo, al mismo tiempo, la mayor cuota del 70.4% de las bitácoras son **diarios personales** que cuentan las vidas de sus autores, bastantes menos están dedicados a filtrar, esto es a comentar contenido externo y a compartir conocimientos. Schmidt & Mayer (2007) manifiestan resultados similares en su estudio de finales de 2005 entre bloggers de habla alemana: los usuarios educandos (alumnos, estudiantes) están sub-

representados entre los escritores de *blog* que se proponen compartir conocimiento (también conocidos como bloggers del conocimiento = c-bloggers); la mayor parte de c-bloggers provienen del contexto laboral [8].

Las razones por las que la gente crea y mantiene *blogs* varían en gran medida. Sin embargo, siempre incluyen la construcción de comunidad y redes sociales entre las motivaciones clave sustentadas a través de los estudios empíricos [9][10]. Además de los obvios *blogs* de grupo, las redes sociales han demostrado existir entre *blogs* individuales pero conectados en red [11].

Sin embargo, al desplegar *blogs* en colaboración, pueden encontrarse muchos obstáculos que aún no se han superado. Para facilitar las "conversaciones de *blog* produc-

ecos de actualización (*update pings*), y sincronizar los cambios.

Mirando más de cerca la **figura 1**, los estándares de interacción actuales parecen concentrarse primariamente en las funcionalidades de publicación y referencia, mientras que las áreas de red activa y pasiva están claramente menos soportadas. Las características de red activa incluso están ausentes por completo.

Para concluir, la falta actual de coherencia conversacional que hemos esbozado antes es el resultado de la ausencia de un soporte para redes activas y de las limitadas facilidades para redes pasivas. Se necesita un estándar de interacción orientado directamente a estas funcionalidades. Sin los avances en infraestructura, la colaboración estrecha puede lograrse únicamente a costa de los usuarios.

3. Especificación de la gestión de la alimentación

El proceso de colaborar vía *blog* puede dividirse en dos peldaños independientes, la ges-

ción de los alimentadores y los canales de comunicación, que incluye la autorización y el intercambio de elementos o colecciones de elementos (la materialización, la transmisión de contenido en sí misma). Esta sección describe el eslabón perdido, una especificación para gestionar las suscripciones de alimentación en un escenario distribuido que complementa los estándares existentes, como se han analizado antes. A esta especificación se le denomina "FeedBack".

Los servicios de agregación están "abusando" ya de la especificación de solicitud de notificación (*pingback*) en la medida en que están usando la llamada derivada *weblog.ping* XML-RPC para informar sobre elementos nuevos y actualizados pero no para informar sobre respuestas a mensajes de *blog* preexistentes. Sin embargo, al mismo tiempo, no hay opciones estandarizadas para informar al sistema sobre la existencia de un alimentador y sobre las actualizaciones para permitir una mejor gestión de la sincronización.

Este documento propone un juego de llama-

das de XML-RPC para transportar la información de gestión de *blog* de un sistema a otro. Es muy ligero, de tal forma que la implementación se hace tan fácil como es posible y las dependencias con otros componentes se reducen al mínimo. El proceso de comunicación completo imita el comportamiento humano y cede el control al usuario siempre que sea posible.

El análisis de rendimiento extensivo de la agregación (*push*) vs. extracción (*pull*) descrito por Deolasee et al. (2001) muestra que para requerimientos de coherencia temporal pequeña, la extracción implica desventajas de rendimiento [14]. La sincronización de datos basada en agregación proporciona ciertas ventajas sobre las basadas en extracción, siendo la más importante la inmediatez. La extracción requiere dos pasos de comunicación, mientras que la agregación mantiene la información del estado y solo envía datos cuando es necesario: preserva la información sobre los intereses de los clientes y agrega solamente la información relevante. Las interacciones de la extracción requieren muchos accesos sin resultado, ya

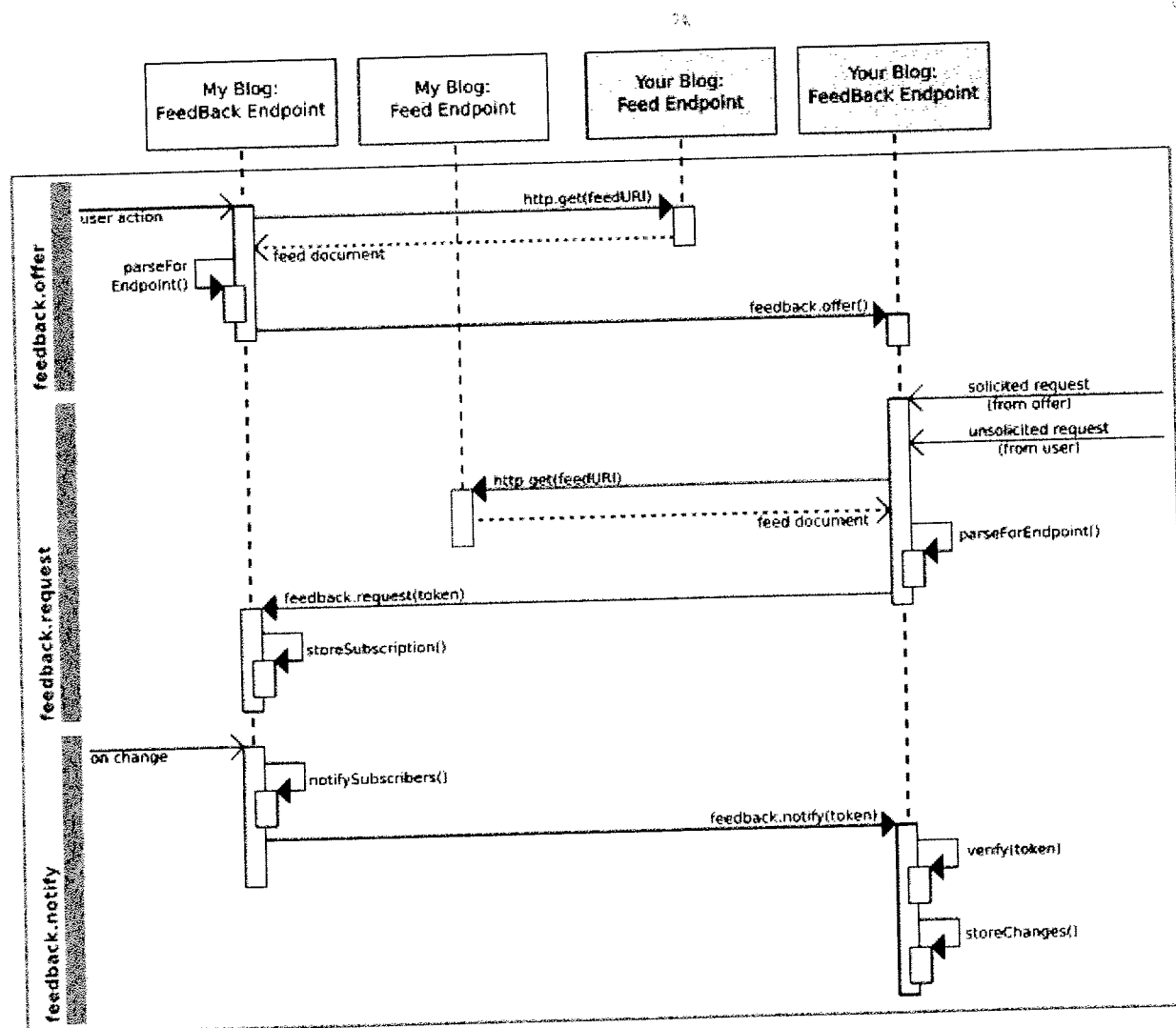


Figura 2. Procesos de comunicación con FeedBack.

que los *blogs* normalmente no cambian con mucha rapidez (sino en momentos distintos). Como consecuencia, la extracción causa una mayor carga de tráfico en la red, especialmente con un gran número de clientes. En un enfoque de agregación, la acción se realiza solo cuando es necesario. Aún en la inmediatez inherente de un enfoque basado en la agregación puede subyacer un gran escollo si se despliega en un sistema grande y distribuido. Dependiendo de la volatilidad de los datos involucrados en un sistema grande y distribuido, así como del número de nodos que requieran sincronización, un enfoque basado en la agregación puede ser afectado por la sobrecarga de las notificaciones pequeñas que se generan y transmiten. Este problema generalmente se resuelve usando un sistema de *buffer* inteligente que combina varias notificaciones de actualización en una ventana de tiempo dada, un enfoque de "agregación con *buffer*".

La especificación *FeedBack* establece cuatro propiedades distintas para los sistemas que la cumplan. Primero, para propósitos de descubrimiento un sistema que admita *FeedBack* ha de ser capaz de devolver una URI a su punto final XML-RPC que admita *FeedBack*, bien usando una cabecera llamada "X-Feedback" o teniendo un elemento *LINK* en cada página que admita *FeedBack*, con el atributo *REL* puesto a "FeedBack". El punto final en cuestión debe soportar los tres métodos siguientes:

- **feedback.offer**: usado para anunciar un alimentador al propietario de un URI particular en el punto final.
- **feedback.request**: usado para registrar las notificaciones de actualización.
- **feedback.notify**: usado para informar sobre la existencia y el contenido de las actualizaciones.

Los sistemas que pretendan comunicar, primero detectan el punto final vía la cabecera *http* del elemento *LINK*. Subsecuentemente, entran en el proceso de comunicación usando los tres métodos mencionados antes. Usando estos cuatro pasos: descubrimiento, oferta, petición y notificación, la especificación proporciona los medios para que las suscripciones de alimentación basadas en el envío automático sean simples y seguras.

4. Proceso de comunicación

En esta sección describimos el intercambio de datos, los pasos de interacción, y las transiciones de estado entre dos sistemas de *blog* y sus usuarios, cuando se comunica información de gestión sobre un alimentador particular y los elementos específicos en este alimentador.

FeedBack puede utilizarse para facilitar dos modos de entrar en la suscripción de alimentación. Las suscripciones pueden solicitarse,

cuando haya habido una oferta oficial de un alimentador por parte de un sistema fuente al sistema destino. Sin embargo, las suscripciones pueden ser también no solicitadas, cuando el sistema de destino envía una petición de actualizaciones de alimentación.

Describimos en forma de ciclo completo el proceso de solicitud de suscripción de notificación y actualización *FeedBack*. Nótese que el primer paso ("offer") es opcional y puede obviarse. El vocabulario de descripción debería resultar familiar a partir de la figura 2, y se refiere a un escenario ficticio de dos *blogs*, "My Blog" que será la fuente de actualizaciones en este escenario, y "Your Blog" que será el receptor de las notificaciones de actualización.

Las llamadas *offer* y *request* ejecutadas entre los dos sistemas son las necesarias para que tal relación se establezca con seguridad. Cada paso contendrá un pequeño bloque de pseudocódigo de fácil comprensión con el fin de clarificar aún más el mecanismo. Dado que la especificación *FeedBack* es un API del lado del servidor, toda la funcionalidad ofrecida en este escenario se accede desde el panel de control de cada sistema respectivo.

Para facilitar la comprensión, la siguiente descripción del proceso de comunicación está fraseada como si uno de los autores de este artículo estuviera dirigiéndose a usted, el lector, directamente.

4.1. Ofrecer la suscripción

Si yo quiero usar *FeedBack* para compartir la alimentación de mi *blog* con usted, entro en el panel de control de My Blog, y le digo que ofrezca a Your Blog una suscripción de la alimentación de My Blog. Hago esto entrando el URI (*Uniform Resource Identifier*) de Your Blog (idealmente cualquier página dinámica perteneciente a Your Blog).

My Blog lee el código fuente del URI dado, para intentar encontrar bien una cabecera *x-feedback HTTP*, o bien un elemento *LINK* que contenga un URI con un punto final XML-RPC válido que soporte los métodos *FeedBack*. Una vez que encuentre un URI válido, procederá a crear una instancia del punto final de Your Blog y hará una llamada *feedback.offer*.

```
xrpc = new xmlrpc("http://your.blog.org/xmlrpc")
xrpc.feedback.offer(
  "http://your.blog.org/uri", 1
  "http://my.blog.org/advertised_feed" 2
)
```

¹ El URI que he introducido en mi panel de control, que My Blog usó para encontrar el punto final XML-RPC de Your Blog.

² El URI para la alimentación en My Blog que quiero compartir con usted.

La alimentación que ofrece My Blog, tiene que adherirse por supuesto a las mismas condiciones que cualquier sistema habilitado para *FeedBack*, y tener la cabecera *X-Feedback HTTP* (o el elemento *LINK* mencionado antes) para suministrar un punto final XML-RPC válido, para interactuar con Your Blog. Al recibir la oferta de suscripción, Your Blog ha de recuperar la alimentación ofrecida, guardar algunos de los detalles del metadato para ayudarle a decidir si se suscribe o no, y guardar el punto final XML-RPC del URI en cuestión para propósitos de interacción posterior en caso de que acepte la oferta.

4.2. Petición de suscripción

Your Blog ha mostrado una oferta de suscripción pendiente desde My Blog. Si decide aceptar esta oferta, Your Blog hará una llamada *feedback.request* al punto final XML-RPC que encuentre en el alimentador ofrecido. La petición le dirá al punto final XML-RPC de My Blog qué canal de alimentación se está solicitando, a donde (a qué punto final XML-RPC) enviar las notificaciones, y Your Blog generará un patrón para identificar con seguridad las notificaciones de actualización futuras a medida que lleguen.

```
xrpc = new xmlrpc("http://my.blog.org/xmlrpc")
xrpc.feedback.request(
  "http://my.blog.org/advertised_feed", 1
  "http://your.blog.org/xmlrpc", 2
  "098f6bcd4621d373cade4e832627b4f6" 3
);
```

¹ En este caso el canal de alimentación solicitado es naturalmente el mismo que ofreció My Blog y que yo quise compartir.

² My Blog no tiene forma de conocer a dónde enviar las notificaciones de actualización si no se le proporciona un punto final XML-RPC de destino.

³ El patrón generado por Your Blog para identificar las notificaciones de actualización entrantes.

En caso de que la *feedback.request* se haga directamente, sin el paso de la oferta, la forma natural de que Your Blog la maneje sería, ver primero si el alimentador que usted está solicitando soporta *FeedBack*. Si no, o si el punto final facilitado no soporta el método *feedback.request*, se recomienda que la implementación *FeedBack* de Your Blog se reduzca a solicitar la extracción de la alimentación a intervalos regulares.

4.3. Notificaciones de actualización

En este punto, Your Blog ha hecho una llamada *feedback.request* a My Blog, y cuando envíe una nueva entrada que debería

aparecer en el alimentador que ha solicitado Your Blog, entonces My Blog enviará una notificación de actualización, mediante el método `feedback.notify`:

```
xrpc = xmlrpc.server("http://  
your.blog.org/xmlrpc")  
xrpc.feedback.notify(  
"http://my.blog.org/post/data.xml", 1  
"098f6bcd4621d373cade4e832627b4f6" 2  
)
```

¹ El parámetro de carga útil debería apuntar a un URI que contenga las actualizaciones de las que Your Blog debería estar advertido.
² El patrón proporcionado originalmente a My Blog por Your Blog, identificando que ésta actualización es deseada.

Your Blog busca el patrón recibido por la notificación de actualización para encontrar la suscripción correspondiente. Si Your Blog soporta diferentes formas de suscripción a las actualizaciones vía FeedBack, debería buscar cómo manejar esta notificación particular basándose en el patrón. Las reacciones posibles podrían incluir volver a recuperar la alimentación original, la inserción, el borrado, o la actualización de las entradas halladas. Esta opción es además simple de implementar, y por tanto puede ser la más común. Otras reacciones pueden incluir la recuperación de los datos apuntados por el URL del parámetro de carga útil (*payload*), para encontrar, por ejemplo, un registro de transacciones de los cambios en la alimentación. En el caso de *blogs*, esto podría ser un alimentador que contenga una información de actualización más detallada, tal y como lo ofrece Atompub [17].

5. Implementación de la referencia

Para demostrar la aplicabilidad del eslabón perdido "FeedBack", se ha desarrollado una implementación de referencia para WordPress, el popular entorno de *blogs*. El conector (*plug-in*) puede descargarse vía SourceForge [18]. Actualmente, el uso de FeedBack se investiga en campo de pruebas dentro del proyecto de investigación iCAMP

con fondos de la UE. Hasta ahora, se han registrado 17 *blogs* usando el conector FeedBack, resultando 68 ofertas que se reflejan en 45 ofertas aceptadas. Los *blogs* recibieron un total de 469 notificaciones mientras que enviaron 504 (incluyendo también las notificaciones a otros sitios).

Se ha creado un servicio de validación para probar el cumplimiento de la especificación [19] dando soporte a los desarrolladores para escribir las nuevas implementaciones.

6. Conclusión y visión general

FeedBack complementa los estándares y las especificaciones existentes en la blogosfera para soportar el trabajo de articulación de la gestión de ofertas, suscripciones, y rutinas de actualización en procesos de aprendizaje colaborativo. Se ha diseñado para que sea simple y robusto y pueda aplicarse a casi cualquier tarea de sincronización de datos donde los sistemas estén sujetos a escenarios de uso dinámicos de software social, la heterogeneidad de los sistemas sea común, y la complejidad no requerida en el nivel de transporte. El trabajo futuro necesita explorar la aplicabilidad de FeedBack a procesos complejos, en cascada y con diferentes formatos de datos con mayor profundidad.

Referencias

- [1] Ravi Kumar, Prabhakar Raghavan, Jasmine Novak, Andrew Tomkins. On the Bursty Evolution of Blogspace, *Proceedings of the WWW2003*, Budapest, Hungary, ACM Press, pp. 568-576.
- [2] Technorati. *About Us*, <<http://technorati.com/about/>>, accedida el 10 de enero de 2008.
- [3] David Gurzick, Wayne G. Lutters. From the Personal to the Profound: Understanding the Blog Life Cycle. En *Proceedings of the CHI 2006*, Montreal, Canada, ACM Press.
- [4] David Sifry. *The State of the Live Web*. Abril 2007, <<http://www.sifry.com/alerts/archives/000493.html>>, accedida el 9 de enero de 2008.
- [5] Michele Tepper. The Rise of Social Software. En *netWorker 7(3)*, ACM Press, 2003, pp. 18-23.
- [6] eLearning Centre. *Products & Services: Blogging Tools*. eLearning Centre, Learning Light, 2006. <<http://www.e-learningcentre.co.uk/eclipse/vendors/>

- [weblogging.htm](#)>, accedida el 5 de julio de 2007.
- [7] Susan Herring, Louis Scheidt, Sabrina Bonus, Elijah Wright. Bridging the Gap: A Genre Analysis of Weblogs. En *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37)*, IEEE Computer Society Press, 2004.
 - [8] Jan Schmidt, Florian Mayer. *Wer nutzt Weblogs für collaborative Lern- und Wissensprozesse?*. En: Dittler, Kindt, Schwarz (Eds.): *Online-Communities als Soziale Systeme*, Waxmann, New York/München/Berlin.
 - [9] Bonnie Nardi, Diane Schiano, Michelle Gumbrecht, Luke Swartz. Why We Blog. En *Communications of the ACM 47(#2)*, ACM Press, 2004, pp. 41-46.
 - [10] Diane Schiano, Bonnie Nardi, Michelle Gumbrecht, Luke Swartz. Blogging by the Rest of Us. En *Proceedings of the CHI 2004*, Vienna, Austria, ACM Press.
 - [11] Alvin Chin, Mark Chignell. A Social Hypertext Model for Finding Communities in Blogs. En *Proceedings of the Hypertext 2006 (HT'06)*, Odense, Denmark, pp. 11-22.
 - [12] Aldo De Moor, Lilia Efimova. An Argumentation Analysis of Weblog Conversations. En Aakhus, Lind (eds.) *Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2004)*, New Brunswick, USA, pp. 197-211.
 - [13] Steve Krause. *When Blogging Goes Bad: A Cautionary Tale about Blogs, Email Lists, Discussion, and Interaction*. 2004. <<http://english.ttu.edu/KAiros/9.1/praxis/krause/index.html>>, accedida el 11 de enero de 2008.
 - [14] P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham, P. Shenoy. *Adaptive Push-Pull: Disseminating Dynamic Web Data*. ACM, 2001. <<http://www10.org/cdrom/papers/pdf/p269.pdf>>. Accedido el 14 de enero de 2008.
 - [15] Heinz Wittenbrink. *Newsfeeds mit RSS und Atom*. Galileo Press, Bonn, 2005. ISBN: 3898425622.
 - [16] Fridolin Wild. *An Interoperability Infrastructure for Distributed Feed Networks*. Deliverable D3.3, iCamp Consortium, 2007.
 - [17] J. Gregorio, B. hOra. The Atom Publishing Protocol. RFC 5023, IETF, 2007. <<http://tools.ietf.org/html/rfc5023>>. Accedido el 14 de enero de 2008.
 - [18] Steinn Sigurdarson, Fridolin Wild, Ahmet Soyulu. *FeedBack Wordpress Plug-in*. 2007. <http://sourceforge.net/project/showfiles.php?group_id=191261&package_id=224627>, accedida el 14 de enero de 2008.
 - [19] Steinn Sigurdarson. *FeedBack validator*. <<http://validator.icamp.eu>>, accedida el 14 de enero de 2008.
 - [20] Tim O'Reilly. *What is Web 2.0*. 2005. <<http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>>, accedida el 17 de marzo de 2008.

¿Estudiante de Ingeniería Técnica o Ingeniería Superior de Informática?

Puedes aprovecharte de las condiciones especiales para hacerte

socio estudiante de ATI

y gozar de los servicios que te ofrece nuestra asociación,

según el acuerdo firmado con la

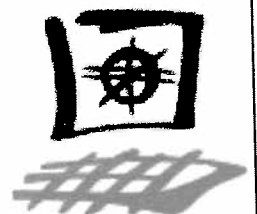
Asociación RITSI

Infórmate en <www.ati.es>

o ponte en contacto con la Secretaría de ATI Madrid
secretmadr@ati.es, teléfono 91 4029391



www.ati.es



www.ritsi.org