

Document downloaded from:

<http://hdl.handle.net/10251/43732>

This paper must be cited as:

Leiva Torres, LA.; Vivó Hernando, RA. (2007). (smt) Herramienta de registro y visualización de mouse tracking en tiempo real para la evaluación de usabilidad en sitios web. *Novática*. (189):53-60.



The final publication is available at

<http://www.ati.es/novatica/indice.html#2000s>

Copyright Asociación de Técnicos de Informática (ATI)

(smt) Herramienta de Registrado y Visualización de Mouse Tracking en Tiempo Real para Evaluación de Usabilidad en Websites

Luis A. Leiva Torres, Roberto Vivó Hernando
luileito@doctor.upv.es, rvivo@dsic.upv.es
Dept. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera, s/n 46022 Valencia (Spain)

Resumen

El mouse tracking en websites empleando tecnologías web estándar es una interesante aunque no evidente herramienta para el análisis de la interfaz de usuario, un gran complemento a los clásicos logs del servidor. Este artículo se centra principalmente en la interacción del usuario a través de los movimientos y clicks de ratón. Se presenta un sistema de mouse tracking que registra, gestiona y reproduce en tiempo real los movimientos de los usuarios en websites, lo cual es un buen punto de partida para modelar un perfil de conductas y tendencias.

Palabras Clave

Mouse tracking, seguimiento del ratón, actividad del ratón, visualización en tiempo real, Interacción Persona-Ordenador, IPO, CHI, interacción web, evaluación de usabilidad, estudio del usuario, tendencias de comportamiento, diseño web.

1. INTRODUCCION

Navegar por la web es una tarea común que desde hace años se ha convertido en nuestro quehacer diario. Existen millones de páginas disponibles 24 horas al día desde cualquier ordenador del mundo ofreciendo todo tipo de información. La web es una economía de la atención, donde la moneda de cambio es el tiempo que el usuario permanece en un sitio [17]. El contenido es el rey. Es el motivo por el que los usuarios vienen a la red. Hay que conseguir que las principales cosas que el usuario quiere hacer se hagan de la forma más sencilla. Por ello es fundamental la estructuración de los contenidos, tanto interna como externamente. La parte externa es lo que más va a influir directamente a la hora de interactuar con los contenidos, mediante un correcto layout; lo que se conoce como la interfaz de usuario.

Hablando estrictamente, la interfaz la componen los dispositivos de entrada y salida y los programas que la hacen funcionar, pero desde un punto de vista más general comprende todo lo que tiene que ver con la experiencia de un usuario y el ordenador, como el entorno de trabajo, la organización en la que trabaja, el entrenamiento que ha tenido, la ayuda que recibe, etc. La Interacción Persona-Ordenador (IPO) es un área de desarrollo reciente, como tantas otras ligadas al campo de los ordenadores, con un carácter marcadamente interdisciplinar y que en los últimos años está viviendo un auge espectacular en sus diversas vertientes. Para estudiar el papel del ser humano en el diseño de sistemas interactivos, además del aspecto informático, hace falta tener en cuenta otras disciplinas. Hay que trabajar los aspectos psicológicos del usuario, la ergonomía del equipamiento, los aspectos sociales, temas de diseño y un largo etcétera que requiere algo más que unos buenos conocimientos de programación.

Así pues, ¿Por qué es tan importante estudiar la interfaz de usuario? Porque es la parte más importante del éxito de una aplicación interactiva. De esta manera, en el presente artículo se pretende

estudiar la usabilidad de toda interfaz actual que exista entre el usuario y la máquina en los nuevos espacios de información, esto es, Internet.

El usuario estándar se desenvuelve en la interfaz de un website mediante el ratón principalmente y, en menor medida, el teclado. Sin duda el ratón es el dispositivo más empleado: está en manos del usuario y él lo decide todo. Todo está a un click de distancia. Una de las principales vías para evaluar la usabilidad de una interfaz informática, no sólo web, consiste en emplear seguidores de retina ocular (eye tracking), debido a su gran precisión y fiabilidad. El hecho es que estos sistemas no están al alcance de cualquier persona -son demasiado caros- y necesitan un calibrado excepcional para funcionar correctamente, si bien es cierto que una vez conseguidas estas dos premisas -la económica y la puesta a punto- se tiene un análisis adecuado y completo sobre la experiencia del usuario en dicha interfaz. Dado el extenso número de webs que existen en la actualidad y el potencial número de visitantes, es interesante evaluar la interfaz de usuario de formas más económicas. Algunos estudios [6,7] sugieren que hay una fuerte correlación entre los movimientos del ojo y el ratón mientras se navega en la red. Por ello, en los nuevos espacios de información es posible implementar un seguimiento del ratón a distintos niveles de granularidad, con el fin de evaluar la usabilidad de un sitio web.

Hemos llamado a nuestro sistema Simple Mouse Tracking (smt), porque es una simple pero potente herramienta para el seguimiento del ratón en páginas webs. Trabaja a un elevado nivel de interacción con este dispositivo, registrando en una capa independiente todos los movimientos realizados por el usuario en tiempo real. Es posible observar cómo ha interactuado la gente con la interfaz del sitio: dónde pulsan, qué seleccionan o cuánto tiempo están leyendo un texto, entre otras tareas.

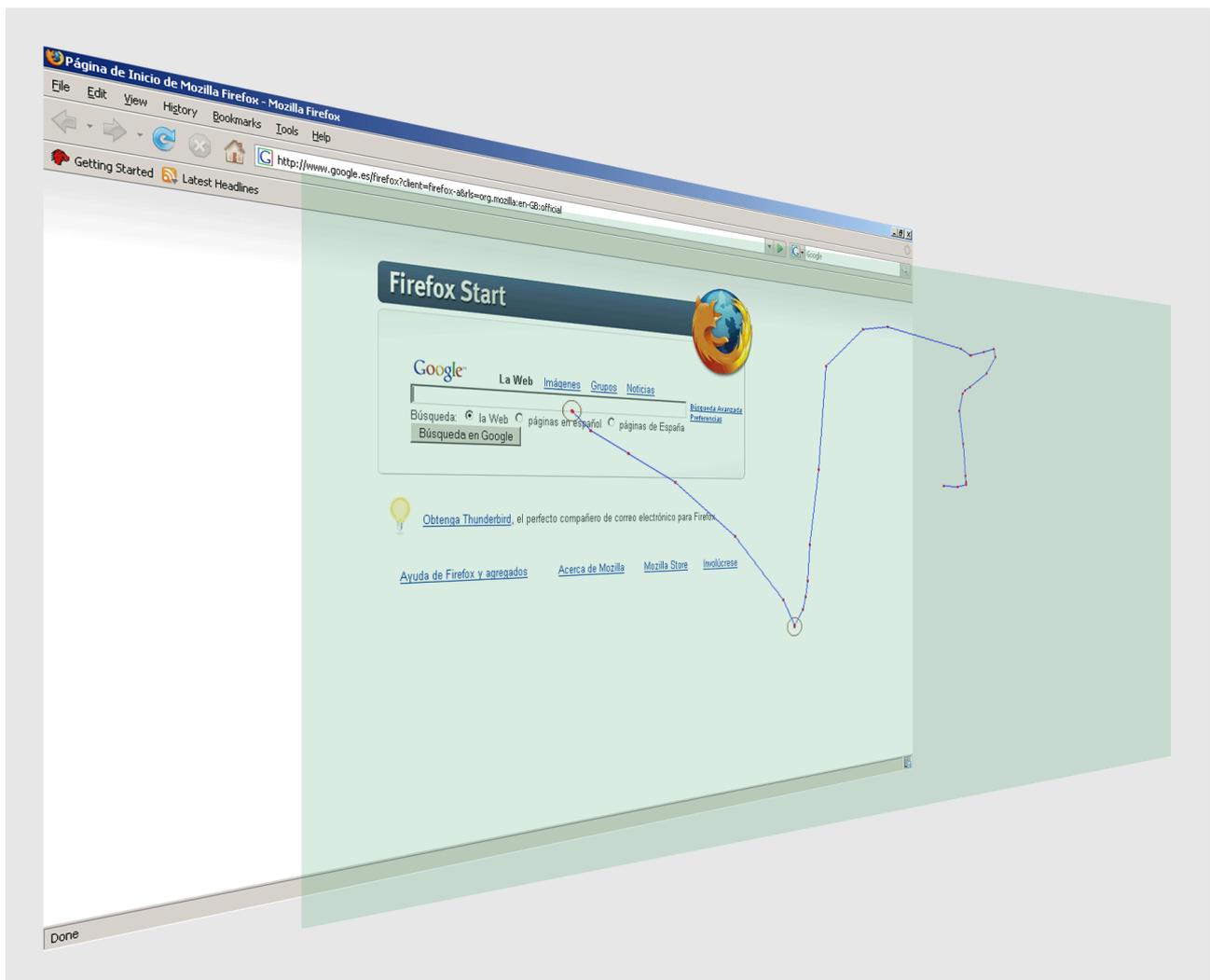


Ilustración 1: El sistema (smt) funciona como una capa de monitorizado por encima de cualquier página web.

Esta herramienta se ha diseñado para ayudar a los administradores y webmasters de un website a realizar rápidos tests de usabilidad y sobre todo a analizar los datos recogidos. Si bien en este artículo se introduce el sistema (smt) como una herramienta para visualización en tiempo real de mouse tracking, su principal objetivo va más allá: analizar las conductas de usuario e inferir un perfil de visitante mediante relaciones psicológicas, sociales y estadísticas.

Este sistema evoluciona progresivamente cada día, en aras de ofrecer una herramienta lo más completa posible, sencilla de implementar y tolerante a fallos, pues recordando la Primera Ley de la Navegación Web de Peter-Paul Koch: *cualquier cosa puede pasar, y normalmente lo hace cuando menos preparado estás para ello*.

2. TRABAJOS RELACIONADOS

Yugo Nakamura [8] fue el primero en experimentar con diversos modos gráficos de mouse tracking e interacciones multiusuario. Ya en 1999 presentó sus fingertracks studies; si bien estos estaban aplicados en páginas flash (SWF) y no HTML, lo cual supone un entorno mucho más controlado que el segundo, que es precisamente el que nos interesa en este artículo. Sin duda su trabajo ha sido y sigue siendo fuente de inspiración para muchos diseñadores y desarrolladores de sitios web.

Existen muchas formas de registrar los movimientos del ratón, pero sin duda la más interesante es la menos obstrusiva, sin que el usuario tenga que instalar nada en su ordenador ni interferir con sus hábitos de navegación. Este requerimiento, lejos de operar en un entorno controlado [8, 9] nos abre un abanico de posibilidades casi infinitas, dado el elevado número de versiones de navegadores web existentes y resoluciones de pantalla, principalmente.

En el caso concreto de los nuevos espacios de información la alternativa más popular ha venido siendo el uso de heatmaps [11], unos mapas digitales que almacenan información de los clicks que el usuario ha realizado en una página web. Esta vía es la más rápida y sencilla de implementar, dado que es un problema ya resuelto, obteniendo resultados similares aunque no tan precisos como los ofrecidos por un eye tracking. Pero esta información puede resultar insuficiente de cara a un análisis más complejo de la interfaz; por ejemplo, no se puede determinar el camino que ha seguido el usuario hasta abandonar la página o dónde ha pulsado.

Por ello han surgido otras alternativas como [1], que mediante un servidor web registran simultáneamente los movimientos del ratón y una marca de tiempo, generando un mapa de ocurrencias con dichas posiciones del ratón. Sin embargo, en este sistema no se tiene en cuenta un vector de dirección, en aras de determinar -nuevamente- de dónde venía y hacia dónde iba el puntero. Así, en [2] se realiza el seguimiento del ratón y se crea un mapa de posiciones, direcciones y clicks, enfatizando en los puntos de entrada y salida a la página. Toda esta información simplemente se empleaba para observar los hábitos de navegación de los usuarios.

Casi paralelamente, ese mismo año se presentó [4], implementado en el proyecto UsaProxy [5]. La metodología es algo similar: mediante una aplicación JavaScript en el cliente se capturan los datos y a través de un servidor proxy se estructuran. Más tarde, mediante una aplicación Java en el servidor, se procesan.

La gran ventaja de los métodos anteriores es que el cliente no percibe que está siendo rastreado, y no necesita instalar ningún plugin en su aplicación cliente, tan solo debe tener un navegador web. Sin embargo, dichos métodos tienen dos inconvenientes, debido a su arquitectura Servidor-Cliente¹: 1) necesitan un servidor proxy para procesar todo el tráfico HTTP del cliente y 2) la creación de los mapas sobre aquellas páginas con mucho scroll implica posteriormente mucho tiempo de generación de dicha imagen, la cual puede llegar a redundar en mayor tiempo de descarga para aquellos clientes que deseen consultar su log.

¹ Por arquitectura Servidor-Cliente nos referimos a que el trabajo de la aplicación recae sobre el servidor, mientras que la presentación de los mismos se ejecuta en el cliente.

A finales de 2006 aparece [10] como un sistema basado en arquitectura Cliente-Servidor², en el cual se solventa el primer inconveniente de los métodos anteriores. Sin embargo, dicha aplicación sólo tiene en cuenta los clicks realizados por el usuario, amén de que sigue compartiendo el segundo inconveniente con sus antecesores.

En 2007 nace [12] como una tecnología para mejorar el mercado de la web 2.0. Este sistema si bien es uno de los más avanzados hasta la fecha, no muestra el recorrido seguido por el usuario, tan solo lo repite sin dejar el rastro del cursor, además de no emplear tecnologías web totalmente estándar. Actualmente se encuentran en fase de desarrollo, mejorando y perfeccionando su herramienta.

Huelga decir que la gran mayoría de los sistemas estudiados hasta la fecha arrojan datos imprecisos al evaluarse en páginas de diagramación líquida y elástica, es decir, de layout variable. Y es que esto es una tarea compleja [14, 16], dado que Internet es un ecosistema muy diverso, donde confluyen variados diseños de páginas con numerosos ordenadores y dispositivos con diferentes navegadores y diferentes resoluciones de pantalla, problema que se acentúa muchísimo más aún en el caso de los dispositivos móviles.

3. CARACTERISTICAS DEL SISTEMA Y OBJETIVOS

Dado que (smt) se trata de una herramienta para cualquiera que esté interesado en cómo interactúan los visitantes en su sitio web, el principal propósito es registrar los movimientos del ratón en tiempo real y en cualquier página web, entendida esta como todo documento en formato HTML. Con este sistema el encargado del sitio, webmaster o administrador, puede repetir y visualizar en tiempo real los movimientos que un determinado usuario ha realizado en sus páginas.

La aplicación más directa es la evaluación de la usabilidad de un sitio web desde cualquier ordenador del mundo, lo cual supone una reducción considerable de costes frente a los actuales tests de evaluación web, los cuales suelen incluir grabación de vídeo in situ y eye tracking, que ya hemos comentado que necesitan una preparación previa del entorno de trabajo: instalación de software, calibrado, horarios de trabajo, anotaciones, etc. Por no decir que gracias a la evaluación online los muestreos son considerablemente mayores, ya que el número de potenciales usuarios que puedan acceder a un sitio web es enorme y más enriquecedor en cuanto a datos reales se refiere: diferentes usuarios con diferentes equipos, diferentes navegadores, diferentes conocimientos informáticos... en definitiva un abanico de posibilidades mucho más amplio que el que puede obtenerse en un laboratorio.

Hay que remarcar que este sistema no pretende ser un sustituto de los actuales test de usabilidad, si no un complemento que además puede funcionar independientemente, enfocado a aquellos profesionales con un menor nivel de exigencia –en cuanto a test de usabilidad se refiere- o que necesitan hacer un análisis rápido de un layout que acaban de publicar online, por ejemplo.

El gran objetivo de partida es que el sistema sea lo más sencillo de implementar, tanto en preparación como en uso. Por ello, no es necesario instalar software adicional ni en el cliente ni en el servidor. Tampoco es necesario emplear una base de datos, aunque esto podría ser un requisito en función de las necesidades de cada webmaster.

Otro gran objetivo es que el sistema propuesto sea capaz de trabajar en cualquier tipo de layout -en términos de diagramación de contenidos- y en cualquier ordenador o máquina cliente, independientemente de su resolución de pantalla y del tamaño de la ventana en la que visualice la web. Para ello, las coordenadas de la trayectoria almacenada (c) son normalizadas de acuerdo a la expresión

$$c^* = c \cdot \delta \tag{ec. 1}$$

² En este caso el navegador del cliente recoge los datos y el servidor solamente los procesa.

Donde δ es el ratio de discrepancia, tanto en horizontal como en vertical, definido por el tamaño de la ventana en la que se realizó la visita (w_2) y el tamaño de la ventana actual (w_1) según

$$\delta = \frac{w_1}{w_2} \quad (\text{ec. 2})$$

Este ratio nos informa precisamente de la correlación que existe entre las dimensiones de la ventana actual -donde se está visualizando la trayectoria- y la original -donde se registró dicha trayectoria-. Por lo tanto, un valor unitario en anchura y altura nos indica que ambas ventanas tienen el mismo tamaño, lo cual sólo ocurre cuando el navegador también es el mismo. Cuando este ratio no es la unidad, lo cual es la situación más frecuente, los valores normalizados varían sensiblemente de los originales. Ello es debido en gran parte al truncamiento de decimales que inevitablemente se produce, en aras de aliviar la carga computacional.

Esta herramienta funciona en cualquier sistema operativo con cualquiera de los navegadores web modernos: Explorer, Firefox, Opera, Safari, Camino o Gecko, entre otros. Dicho sistema no interfiere con otros scripts que existan en la página o que se incluyan posteriormente, lo cual es una característica deseable y por supuesto requerida.

Asimismo, se puede configurar fácilmente la precisión del registrado y los directorios donde almacenar los informes de cada sesión, así como el tiempo de registrado. Es conveniente establecer un timeout, puesto que el ratón es monitorizado constantemente hasta abandonar la página y pueden darse ocasiones en las que sea inabordable el tratamiento de los datos: por ejemplo si el usuario abre su navegador y abandona la habitación.

Una característica muy interesante es que dicha herramienta comienza a monitorizar la actividad del ratón cuando el Modelo de Objetos del Documento (DOM) está listo, por lo que no es necesario esperar a que todo el contenido binario se descargue [15], como imágenes o contenido embebido (películas flash o quicktime, por ejemplo). El DOM³ es una forma de representar los elementos de un documento estructurado, tal como una página web HTML o un documento XML, como objetos que tienen sus propios métodos y propiedades, propuesto por el World Wide Web Consortium (W3C).

Otra característica destacable es que se genera un log en formato XML por cada usuario y página visitada, de tal forma que se facilitan notablemente las tareas de postproceso, ya que la gran mayoría de los lenguajes de programación para la web disponen de APIs para acceder a documentos en dicho formato. De esta forma, es posible generar estadísticas a partir de los datos arrojados, las cuales combinadas con cualquier otra herramienta de análisis, como Urchin o Google analytics, hacen de (smt) un eficiente sistema de evaluación.

Por último, y no por ello menos importante, el principal propósito de esta aplicación consiste en analizar las conductas de los usuarios e inferir un perfil de visitante mediante relaciones psicológicas, sociales y estadísticas. Este es un estadio que deberá alcanzarse cuando se disponga de un número de muestras suficientemente representativo. Como orden de magnitud se proponen entre 400 y 500 logs procesados, dadas las posibles anomalías que pudieran presentarse, o entre 20 o 30 logs para cada usuario. Si bien este segundo caso sería el preferible, de cara a personalizar una página en función de las características personales de cada visitante (por ejemplo el posicionado y la distribución de elementos, tamaño de los textos, etc.), en la realidad es prácticamente imposible conseguirlo si no es mediante un sistema de autenticación de usuarios o realizando pruebas en laboratorio, siendo este último un escenario muy restringido y poco extrapolable a situaciones reales.

4. ARQUITECTURA DEL SISTEMA

Esta es una herramienta Open Source de arquitectura cliente-servidor para webmasters,

³ En efecto, el DOM es una API para acceder, añadir y cambiar dinámicamente contenido estructurado en dichos documentos.

desarrolladores e incluso diseñadores de sitios web. La aplicación propuesta consta básicamente de dos partes: la etapa de registrado y la etapa de visualización. En la primera se ejecuta una aplicación en el browser del cliente (un fichero JavaScript) y una aplicación en el servidor (un fichero PHP) se encarga de procesar los datos, ambos archivos funcionando de forma transparente para el usuario. El núcleo de (smt) para la grabación en tiempo real reside en el almacenamiento de las coordenadas del ratón a intervalos regulares de tiempo, medidos en fotogramas por segundo –una unidad de tiempo amigable para la mayoría de usuarios–.

Para visualizar los resultados se emplea únicamente un fichero JavaScript, el cual se sirve a su vez de la potente High Performance JavaScript Vector Graphics Library [13]. Al trabajar con gráficos vectoriales, nuestra herramienta de visualización supone un tiempo de descarga mínimo, ya que los únicos datos a cargar es la misma página que visitó el usuario junto con sus movimientos de ratón.

El funcionamiento de (smt) es simple, tal y como se observa en la figura 2. La generación de la página web con los datos del registrado se realiza en el servidor y en el mismo momento en que el usuario abandona dicha página, con lo que las revisiones de los movimientos están disponibles desde ese mismo instante.

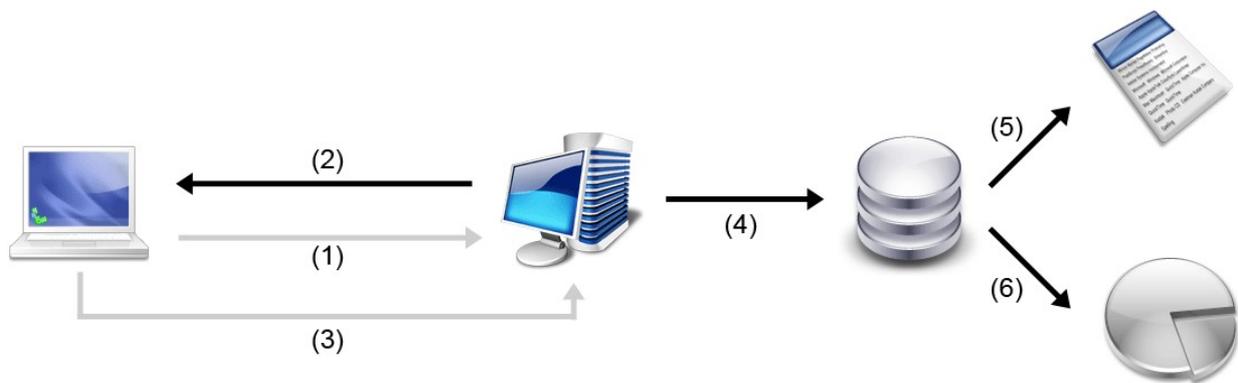


Ilustración 2: Arquitectura del sistema. (1) El cliente accede al servidor web y este (2) le devuelve la página con la herramienta de monitorizado. Al abandonar la página (3) los datos son enviados de forma asíncrona al servidor, donde (4) son procesados y almacenados. Se crea (5) una copia de la página visitada con la capa de tracking y (6) un fichero XML como log de los datos del visitante.

4.1. Preparación

Para comenzar a utilizar (smt) no hay que efectuar ningún cambio en la máquina cliente ni en el servidor. Tan solo se necesita un servidor de páginas web con soporte para PHP –prácticamente todos los servidores web tradicionales ofrecen esta posibilidad–, donde ubicar el archivo encargado de procesar los logs de las páginas visitadas, por ejemplo en un directorio cgi-bin o similar. En dicho archivo PHP es posible configurar los paths, relativos o absolutos, a los logs en HTML y XML respectivamente.

El webmaster debe incluir simplemente una línea de código JavaScript⁴ en aquellas páginas que desea monitorizar, después de la etiqueta <body>:

⁴ Desde Abril de 2006 se recomienda insertar código JavaScript externo en una página web mediante el Media Type `application/javascript` en lugar de `text/javascript`, pero esta última es, por motivos históricos, la forma que reconocen todos los navegadores web.

```
<script type="text/javascript" src="path.to.js/smt_record.js"></script>
```

Dichas páginas sólo se ven modificadas por esta línea de código, y, lo que es más importante, este script externo se almacena en la caché del navegador, luego no tiene que recargarse cada vez que se visita otra página del sitio que también vaya a ser monitorizada.

Un programa JavaScript sólo tiene acceso a la página donde se encuentra insertado, según su modelo de seguridad SandBox. Precisamente por eso se deben modificar manualmente aquellas páginas que se desea monitorizar. Esto puede realizarse de manera automática, y sin emplear un servidor proxy, pero hemos decidido hacerlo así de cara a un análisis preliminar de la herramienta y de esta forma centrarnos en el desarrollo de la misma de manera más efectiva. Una solución eficiente es escribir un programa que se ejecute una sola vez en el servidor y busque recursivamente todos los documentos HTML, PHP o ASP del sitio, por ejemplo, insertando dicha línea de código JavaScript tras la etiqueta `<body>`. Pero este no es ahora el propósito de nuestro trabajo.

4.2. Registrado

Un usuario se conecta al servidor web y accede a una de las páginas donde se ha incluido el registrado del ratón. Realiza cualquier tarea y al cabo de cierto tiempo abandona la página, bien pulsando un enlace que le lleve a otro documento HTML u otro sitio web, o incluso cerrando la ventana del navegador. En el momento en que se abandona la página, los datos son enviados mediante AJAX al servidor, realizando una conexión asíncrona que no necesita refrescar la ventana actual, con lo que la experiencia de usuario no se ve alterada lo más mínimo y por supuesto no se tiene la sensación de haber sido monitorizado. En el servidor, un fichero PHP se encarga por un lado de crear una copia de la página visitada en el servidor incluyendo la capa de mouse tracking, y por otro lado de generar un log en formato XML con los datos más relevantes de la visita: URL, título de la página, navegador del cliente y sistema operativo, resolución de pantalla, tamaño de ventana con la que ha visitado la web, tiempo de tracking y por supuesto las coordenadas de la trayectoria y clicks realizados en cada instante de tiempo, las cuales son almacenadas en sendos arrays.

4.3. Visualización

El webmaster accede a la carpeta de logs en formato HTML, que no son más que copias exactas de los archivos originales que llevan incluida una capa de monitorizado con los datos correspondientes a cada visita realizada. Al abrir con su navegador uno de estos ficheros, se reproduce exactamente el camino seguido por el usuario hasta abandonar la página. Hemos decidido generar una copia de los archivos originales con la capa de mouse tracking para que así estos puedan ser consultados tanto online como offline, para posteriores estudios de otra índole. Pero no descartamos la idea de emplear en un futuro una base de datos tipo MySQL.

4.4. Inferencia

Una vez recolectados los datos relativos a una página se generan las correspondientes estadísticas, contemplando dos tipos de datos: concretos y abstractos. Los del primer tipo son los que se almacenan en el log XML. Muestran información detallada a partir de los datos de dicho log XML, como por ejemplo cuántas veces se ha pulsado el ratón y en qué instante se ha hecho, si se ha seleccionado algo o el tiempo de permanencia en dicha página, entre otros. Los del segundo tipo son datos que informan sobre la destreza del usuario, cómo se desenvuelve en la interfaz de la web, si es o no impaciente, etc.

Con estos datos y un número de muestras suficientemente representativo es posible inferir las conductas de los usuarios en cada página, creando así un perfil personalizado de cada navegante o uno común a todos ellos. De todos modos, este estadio aun no se ha completado por la falta de suficientes logs como para llegar a conclusiones relevantes.

4.5. Inconvenientes

La mayor desventaja de este sistema es que necesita ejecutar código JavaScript en el cliente, y en la realidad existe una minoría de internautas que navegan por la web con JavaScript deshabilitado. Pero estos representan un porcentaje demasiado bajo como para desestimar la viabilidad de este sistema, siendo la mayor parte de estos los robots que utilizan los buscadores para indexar las páginas.

Un inconveniente menor es que, dado que la etapa de visualización es directamente proporcional a la velocidad del registrado, en visitas de mucho tiempo de duración puede hacerse pesado repetir los movimientos de principio a fin, sobre todo si estos fueron realizados sin ninguna finalidad aparente (por ejemplo, jugueteando por la página). Por ello se establece un tiempo máximo de registrado a modo de timeout, configurable por el webmaster o encargado del sitio web. Es posible determinar estas situaciones estocásticas modelando un perfil de conductas y comportamientos, que no es sino uno de los principales objetivos posteriores a este trabajo.

5. ANALISIS DE RESULTADOS

Este sistema funciona sin problemas en cualquier página de naturaleza HTML. La realidad es que actualmente existen webs con diverso contenido incrustado: animaciones Flash, películas Quicktime o Real Audio, applets de Java, etc. Dada la naturaleza de estos objetos, para poder realizar el seguimiento del ratón sobre ellos es necesario incluir el algoritmo de registrado en el propio objeto, lo cual no es posible en algunos casos, dependiendo del reproductor del contenido.

En la siguiente imagen se muestra la captura de un navegador, a modo de ejemplo, de una visita realizada a una copia⁵ de un portal de vídeos mediante el navegador Firefox 2.

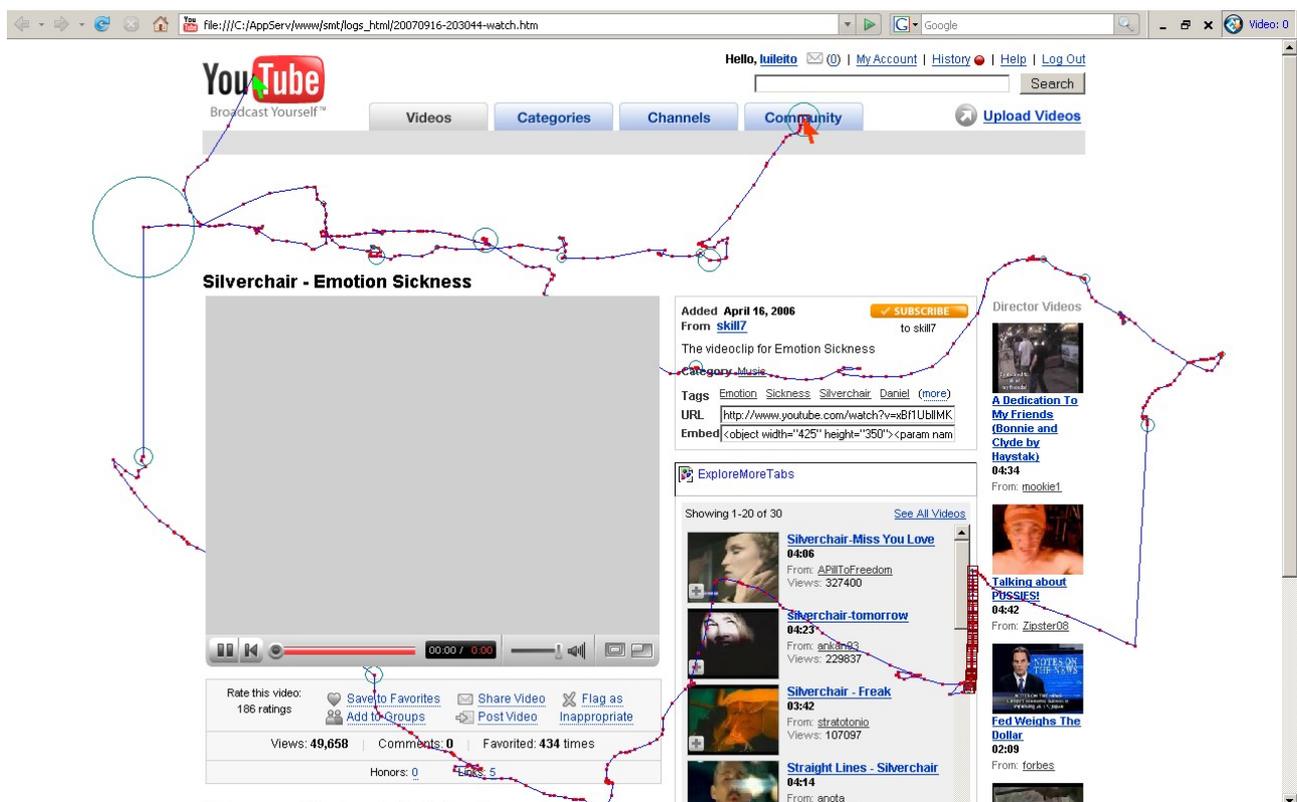


Ilustración 3: Visualización de un recorrido realizado en una web con objetos embebidos.

⁵ En el momento de realizar los análisis, para agilizar el proceso, se ha trabajado en modo local. De ahí que en el log XML se lea como URL `http://localhost/...`

Como se observa, el contenido embebido (en este caso se trata de una película Flash, en formato SWF) se renderiza en un nivel superior al del contenido HTML por defecto en todos los navegadores web, y por ello no se puede observar el camino seguido sobre el mismo. Pero esto es una característica intrínseca del lenguaje HTML. Y es más, el propósito de nuestro sistema es registrar en tiempo real los movimientos del usuario e inferir las conductas de comportamiento. Por ello no supone un problema tan importante como para desestimar su uso.

Un resultado directamente observable es que los usuarios más experimentados (bien porque conocen el website, bien por sus conocimientos informáticos) muestran trayectorias más uniformes y rectilíneas, con un espaciado entre puntos de registro mucho menos denso. En la figura 4 se ilustra esta idea.

El uso de las barras de scroll se puede apreciar fácilmente, ya que el ratón se ubica en el margen derecho de la ventana y la trayectoria es prácticamente rectilínea. Si se emplea la rueda del ratón, la trayectoria es una recta perfecta, tal y como se observa en la figura 5.

Otra característica fácilmente observable es la selección de texto: el botón del ratón es pulsado y se libera cuando la selección del texto ha finalizado. En la figura 6 se recoge el citado concepto.

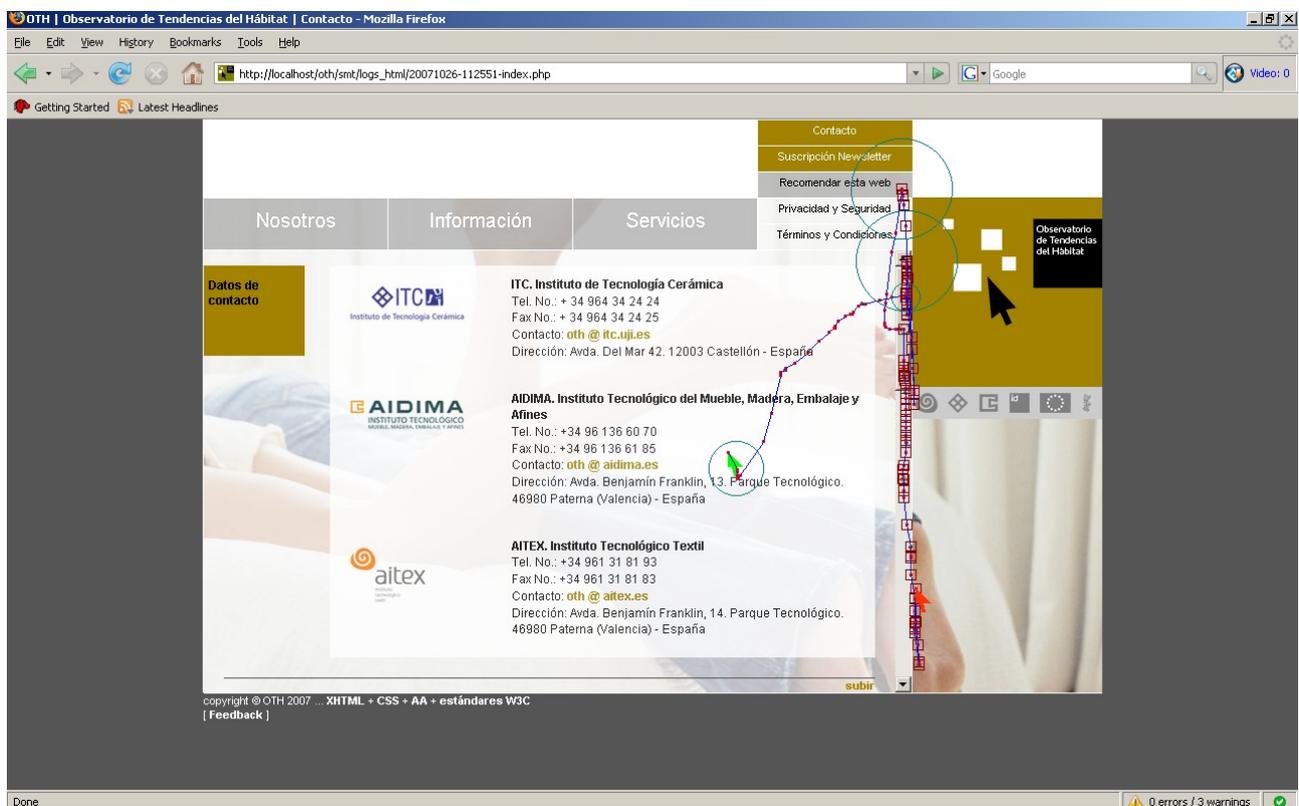


Ilustración 4: Uso de las barras de Scroll, indicando la última posición del ratón antes de abandonar la página. Nota: el último click se realizó sobre el botón, pero esta web utiliza un frame inferior de 70px, correspondientes al offset que se observa en esta imagen.

En la figura 7 se muestran cuatro niveles diferentes de precisión de registrado, efectuados simultáneamente sobre la misma visita a una página. Cabe resaltar que el nivel de precisión es variable y totalmente configurable, medido éste en frames por segundo (fps). Está claro que en el medio está la virtud: imponiendo un número muy bajo de fps no obtendremos una trayectoria similar a la original, mientras que registrando los movimientos a un número elevado de fps podríamos sobrecargar nuestra máquina a la hora de visualizarlos, suponiendo elevados tiempos de monitorizado.

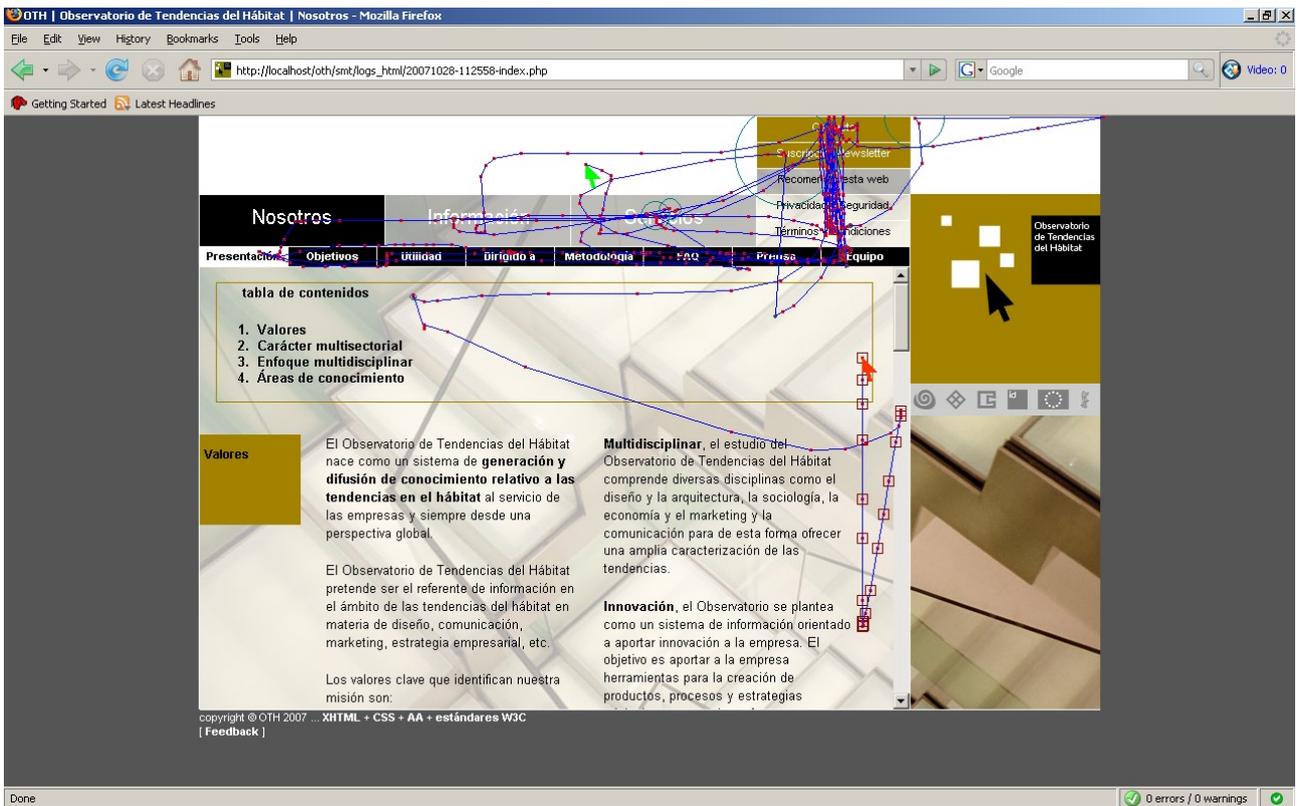


Ilustración 5: Uso de la rueda del ratón en una página, ya que la trayectoria entre los dos puntos de registro consecutivos es perfectamente rectilínea.



Ilustración 6: Selección de texto. Como se observa, se ha realizado desde la esquina superior izquierda hasta la esquina inferior derecha del texto de Contacto, luego deducimos que todo el texto ha sido seleccionado.

Por supuesto, es posible realizar el registro a pocos fps y posteriormente hacer una interpolación para refinar el modelo del camino seguido [3].

En cualquier caso, en un futuro inminente presentaremos las inferencias realizadas mediante esta herramienta de adquisición, el propósito más profundo de este sistema.

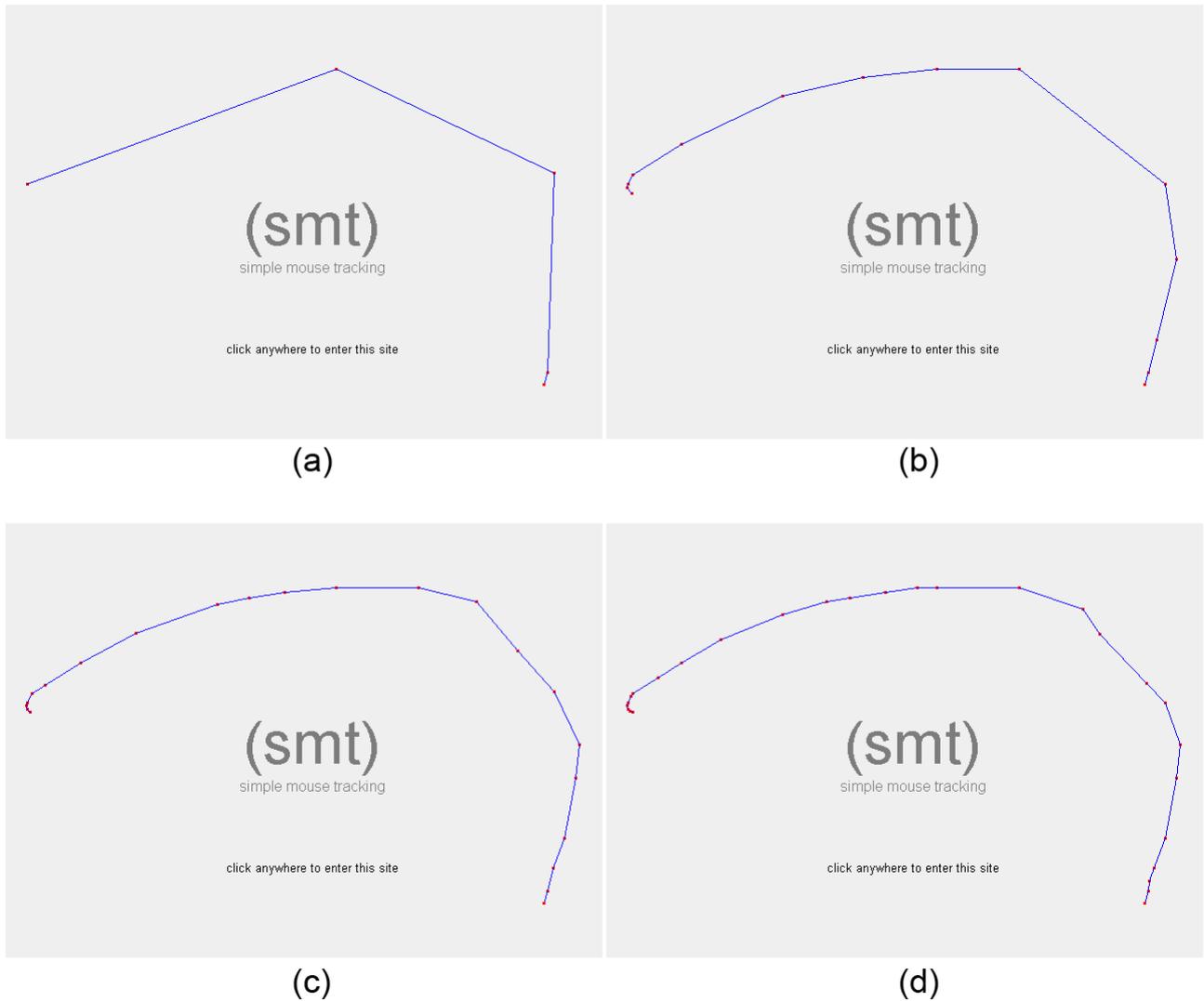


Ilustración 7: Diferentes precisiones de registrado para un mismo movimiento: 5 fps (a), 15 fps (b), 25 fps (c) y 35 fps (d).

Este es el log correspondiente a la visita que aparece en la figura 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<log>
<url>http://localhost/smt/google_record.htm</url>
<title>Mozilla Firefox Start Page</title>
<client browser="Firefox" version="2" os="Windows"></client>
<screen width="1280" height="800"></screen>
<document width="1280" height="669"></document>
<trackingTime>1.922</trackingTime>
<fps>24</fps>
<coordsX><![CDATA[283,272,238,266,299,325,316, ... ,639,637,633,629,626,625]]></coordsX>
```

```
<coordsY><![CDATA[145,174,310,389,394,360,330, ... ,367,363,360,355,353,353]]></coordsY>  
<clicksX><![CDATA[0,0,0,0,0,0,0,0,0,0,0,0,0, ... ,0,0,0,0,0,0,0,0,625,625]]></clicksX>  
<clicksY><![CDATA[0,0,0,0,0,0,0,0,0,0,0,0,0, ... ,0,0,0,0,0,0,0,0,353,353]]></clicksY>  
<clicksTime><![CDATA[1806]]></clicksTime>  
</log>
```

6. TRABAJO FUTURO

Los intentos más ambiciosos de la automatización en los tests de usabilidad provienen de la comunidad de la Inteligencia Artificial. La idea es sustituir a los usuarios por agentes que modelen los procesos perceptivos y cognitivos de los usuarios reales. Estas aproximaciones son tremendamente irresistibles e interesantes pero condenadas a fallar debido a que hasta ahora no ha sido posible generar verdaderos agentes autónomos que actúen como humanos [18]. Dicho intentos han sido extremadamente costosos, demasiado simplistas y totalmente específicos para una tarea o aplicación. Por ello hay que buscar modelos más modestos para registrar, analizar e interpretar las interfaces de usuarios reales.

Como ya hemos comentado en el presente artículo, la siguiente etapa de este trabajo consiste en inferir las conductas de los usuarios, creando así un perfil personalizado para cada uno o para cada página. Para ello es necesario un numero representativo de muestras a analizar, véase entre 400 y 500 logs, dadas las posibles anomalías que pudieran presentarse, o entre 20 o 30 logs para cada usuario.

Otra tarea pendiente es implementar el sistema (smt) para que pueda monitorizar cualquier página desde el servidor donde se encuentre instalado, empleando la tecnología web actual y manteniendo la filosofía “keep it simple”. Sería posible, por ejemplo, grabar en nuestro servidor a un usuario accediendo a otros servidores web.

En el momento de escribir este artículo el sistema (smt) funciona sin problemas en páginas HTML bien formadas, tanto estáticas como dinámicas –en las que el código se ejecuta en el servidor y son devueltas al cliente ya procesadas–. Para este último caso la consulta online es más sencilla de implementar, pero queremos poder disponer de los resultados visuales del mouse tracking en modo offline.

7. CONCLUSIONES

Se ha presentado un sistema para el análisis en tiempo real de los movimientos del ratón en páginas webs, aplicándolo a través de una herramienta visual para la posterior repetición de los mismos.

Este sistema supone un valor añadido para evaluar tanto la usabilidad de un sitio web como la experiencia de usuario, al margen de si es lícito o no el registrar sus movimientos. La herramienta provee la posibilidad de avisar a los usuarios mediante un mensaje de alerta, pidiéndoles permiso para capturar los datos del ratón en el caso de que así lo deseen. Esto podría influir en los resultados, dado que cabe la posibilidad de que algunos usuarios realizaran movimientos intencionadamente poco naturales. De esta forma, en última instancia queda a elección del encargado del sitio web el informar a sus visitantes de que sus movimientos van a ser grabados, así como sobre el tratamiento posterior que se va a hacer con los datos. Esta herramienta ha sido concebida como una ayuda, no como un instrumento de abuso. Por ello el registrado se realiza por defecto de forma totalmente transparente e imperceptible para el usuario y de forma anónima, sin alterar sus hábitos de navegación, y por supuesto sin interferir con otros scripts de la página.

Y este es sólo el punto de partida. Las aplicaciones derivadas son tantas como permita la imaginación: webs autoadaptables en función de los movimientos de los usuarios, ayudas contextuales, soporte técnico en tiempo real o reconocimiento gestural de usuarios por citar solamente algunas.

Reconocimientos

La herramienta de visualización se apoya en la High Performance JavaScript Vector Graphics Library [13] desarrollada por Walter Zorn en 2006.

Existe una dirección permanente donde descargar la herramienta (smt) y comentarla públicamente: smt.speedzinemedia.com.

La ilustración 1 corresponde a la página de inicio del buscador Google.

La ilustración 3 corresponde al portal de vídeos www.youtube.com.

Las ilustraciones 4 y 5 pertenecen a la farmacia virtual www.farmaciadelaconcepcion.com.

La ilustración 6 es al sitio web www.codigojoven.com.

Referencias

- [1] Mueller, F., Lockerd, A. **Cheese: Tracking Mouse Movement Activity on Websites, a Tool for User Modeling**. In Proceedings of the Conference on Human Factors in Computing Systems CHI 2001, extended abstracts on Human factors in computing systems, Seattle, Washington, USA, April 2001.
- [2] Arroyo, E., Selker, T. and Wei, W. **Teaching User Interface Design using a web-based Usability Tool**. In Proceedings of the Conference on Human Factors in Computing Systems CHI 2006.
- [3] Huang, W. **Development of Web-browser Based Thin-client Low Bandwidth Workspace Sharing System with Real Time Display of Remote Mouse Movement**. Thesis. Division of Research and Advanced Studies. Cincinnati, 2005.
- [4] Atterer, R., Wnuk, M. and Schmidt, A. **Knowing the User's Every Move - User Activity Tracking for Website Usability Evaluation and Implicit Interaction**. In Proceedings of The Fifteenth International World Wide Web Conference (WWW2006). Edinburgh, 2006.
- [5] Wnuk, M. **Usability Proxy for Websites**. Thesis. Media Informatics Group. Munich, 2005.
- [6] Chen, M., Anderson, J. R., and Sohn, M. **What Can a Mouse Cursor Tell Us More? Correlation of Eye/mouse Movements on Web Browsing**. Ext. Abstracts CHI 2001, ACM Press (2001).
- [7] Pan, B., Hembrooke, H., Gay, G., Granka, L., Feusner, M. and Newman, J. **The Determinants of Web Page Viewing Behavior: An Eye Tracking Study**. Proc. ETRA (2004).
- [8] Nakamura, Y. **FingerTracks Studies**. Tokyo, 2001.
- [9] Allardice, S. **WPF/E Mouse Track**. San Francisco, 2006.
- [10] Pardo, D. **The definitive heatmap**. Corunna, 2006.
- [11] De Gan, J. **Track mouse movements and clicks**. New York, 2006.
- [12] Nitobi Inc. **RobotReplay - The Next Generation of Web Analytics**. Vancouver, 2007.

- [13] Zorn, W. **High Performance JavaScript Vector Graphics Library**. Munich, 2006.
- [14] P. P. Koch. **Mission Impossible - mouse position**. Amsterdam, 2002.
- [15] Edwards, D., Miller, M. and Resig, J. **The widow.onload problem**. San Francisco, 2006.
- [16] P. P. Koch. **Cross-browser detection**. Amsterdam, 2006.
- [17] Nielsen, J. **Designing web usability: The Practice of Simplicity**. New Riders Publishing. 2nd Edition. ISBN: 1-56205-810-X. Indianapolis, 1999.
- [18] K. L. Norman, E. Panizzi. **Levels of Automation and User Participation in Usability Testing**. Interacting with Computers, vol 18. New York, 2006.