

Herramienta de evaluación de accesibilidad de contenidos y aplicaciones para terminales móviles

Juan Carlos Yelmo García, Yod Samuel Martín García, Beatriz San Miguel González, Ignacio Albornoz Fernández.

Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid.
Escuela Técnica Superior de Ingenieros de Telecomunicación. Ciudad Universitaria, s/n. 28040 Madrid.

Teléfono: 91 336 6830. Fax: 91 336 7333.

Correo Electrónico: {jcyelmo, samuelm, smiguel, ignacio}@dit.upm.es

Resumen

La escasez de herramientas de evaluación automática de accesibilidad para servicios y aplicaciones para terminales móviles hace que, en ocasiones, los servicios móviles no cumplan requisitos de accesibilidad electrónica. Se presenta aquí un prototipo de herramienta que intenta suplir dicha carencia, destinada a evaluar la accesibilidad de contenidos web –definidos mediante WML (Wireless Markup Language) y XHTML-MP (XHTML-Mobile Profile)–, así como de aplicaciones de usuario que ejecuten sobre la plataforma S60 de Nokia. Los criterios de evaluación aplicados por la herramienta se basan en la transposición de distintas familias de pautas de accesibilidad internacionalmente reconocidas, redactadas por consorcios industriales (World Wide Web Consortium) o por Administraciones Públicas (estadounidense e irlandesa). Los resultados de efectuar una evaluación se ofrecen al usuario en forma de informe navegable y se almacenan en un formato estándar de facto en la representación de informes de accesibilidad, denominado EARL (Evaluation and Report Language).

1. Introducción

El concepto de accesibilidad electrónica (*eAccessibility*) prescribe el acceso equitativo a la información con independencia del contexto de uso –capacidades, requisitos y preferencias del usuario; tareas que éste desempeña y plataforma que utiliza–. La accesibilidad se plantea tradicionalmente en relación a grupos cuya participación en la Sociedad de la Información se encuentra en riesgo, como personas mayores y discapacitadas, aunque beneficia a todos los usuarios potenciales. Además, está cobrando un papel cada vez más relevante tanto en la legislación como en el mercado de servicios y aplicaciones, así como en la reputación corporativa de los distintos agentes involucrados en la prestación de servicios telemáticos.

A pesar de esta importancia creciente de la accesibilidad, existen tecnologías donde las posibles modalidades de interacción con el usuario se limitan con frecuencia, lo que dificulta la accesibilidad en múltiples contextos de uso. Las aplicaciones y contenidos para terminales móviles, aun siendo éstos inherentemente multimodales, no siempre tienen en cuenta criterios de accesibilidad durante su desarrollo, reduciendo la interacción con el usuario a una sola modalidad de forma obligada. Esto se debe, en parte, a que se trata de un campo novedoso en donde aún no existen herramientas maduras encargadas de evaluar la accesibilidad de los desarrollos. En este sentido, existen proyectos de investigación para facilitar la accesibilidad en servicios de Redes de Próxima Generación que

incluyen pautas de accesibilidad de servicios móviles [1], si bien se no se dirigen a la evaluación de aplicaciones software, sino al terminal en sí (es decir, al hardware y a las utilidades empotradas por defecto).

Para paliar dicha carencia, el trabajo aquí presentado ofrece una contribución para mejorar la accesibilidad de los servicios y las aplicaciones para terminales móviles, a partir de una herramienta que pueda evaluar su accesibilidad y cuyo empleo permita conocer el grado de satisfacción de distintos requisitos concernientes. En las siguientes secciones se presentan las tecnologías en las que se apoya la herramienta, así como las características de ésta: la sección 2 introduce el escenario habitual en el que se despliegan los servicios y aplicaciones para terminales móviles y muestra la inclusión de la herramienta desarrollada en dicho escenario; en la sección 3 se presenta un estado del arte de la evaluación de la accesibilidad software; la sección 4 expone la influencia en los procesos de negocio de la herramienta desarrollada y su especificación en forma de casos de uso; en la sección 5 se presentan la arquitectura y el diseño de la herramienta de evaluación; finalmente, la sección 6 muestra una síntesis de los resultados, junto con posibles líneas de extensión de la herramienta.

2. Contexto de despliegue

2.1. Marco de ejecución de servicios para terminales móviles

En un servicio para terminales móviles se pueden distinguir habitualmente distintos elementos software y hardware que toman parte en el despliegue, según presenta la Fig. 1.

En primer lugar, encontramos **aplicaciones software** que se despliegan en **nodos terminales** controlados directamente por el usuario, por lo que son las responsables de manejar los elementos de interfaz de usuario, cuyas características pueden variar entre terminales con diferentes prestaciones. En algunos casos, las aplicaciones del terminal utilizadas en un servicio serán aplicaciones creadas y desplegadas *ad hoc*, mientras que, en otros, el acceso al servicio se logrará mediante agentes de usuario genéricos como un navegador web o un visualizador de contenidos multimedia. Las plataformas de ejecución de estas aplicaciones en los terminales conforman una familia heterogénea de sistemas operativos y entornos de ejecución, destacando las plataformas para el sistema operativo Symbian OS (UIQ de Ericsson; Series 40, S60 y Series 80 de Nokia), el sistema operativo Windows Mobile de Microsoft y las distribuciones de Linux para dispositivos móviles. A estas opciones hay que añadir las distintas configuraciones y perfiles de la plataforma Java para terminales móviles, que pueden ejecutarse sobre distintos sistemas operativos.

En segundo lugar, encontramos **aplicaciones remotas** que se ejecutan en **servidores de aplicaciones** y que interactúan, por un lado, con los terminales de usuario y, por otro, con otros elementos del proveedor de servicios o de terceros (bases de datos, capacidades de red, componentes de servicio, etc.) de forma transparente para el usuario final. Las tecnologías de soporte a estas aplicaciones de servidor pueden ser, en general, cualquiera de las habituales en servidores de aplicaciones (Java Enterprise Edition, .NET Framework) y en entornos de ejecución de servicios (OSA/Parlay, JAIN SLEE, etc.).

En tercer lugar, entre ambos aparecen los **contenidos** que las aplicaciones de servidor se

encargan de servir a los programas residentes en el terminal de usuario, y que éstos deberán presentar. Estos contenidos pueden ser estáticos o pueden haber sido generados dinámicamente por la lógica de las aplicaciones de servidor. Los contenidos se pueden definir en distintos formatos, en función del tipo de información que representan y del agente de usuario del terminal al que vayan destinados para su representación. El caso más típico es el de los contenidos web, que suelen venir representados en lenguajes XML inspirados en XHTML, como es el caso de WML (Wireless Markup Language) [2], XHTML Basic [3] o XHTML Mobile Profile [4]. Habitualmente, el desarrollo de todos estos componentes se lleva a cabo en plataformas tipo PC, empleando **entornos de desarrollo** que incluyan capacidades de emulación de los distintos elementos software y hardware del entorno de despliegue.

2.2. Papel de la herramienta de evaluación y ámbito de aplicación

En el proyecto aquí presentado, se ha desarrollado el prototipo de una herramienta de evaluación de accesibilidad de servicios para terminales móviles, con el objetivo de facilitar la evaluación de la accesibilidad a los agentes implicados en la creación y puesta en marcha de un servicio (analistas, desarrolladores, evaluadores) y que puedan desear llevar a cabo dicha evaluación. El destino de la herramienta implica que no se ejecuta en el entorno de despliegue de los servicios (terminales móviles y servidores) sino en el entorno de desarrollo (ordenadores de propósito general).

De entre todos los elementos software anteriormente presentados, el prototipo va dirigido a aquellos que tienen incidencia directa en la interfaz de usuario: las aplicaciones locales del terminal y los contenidos que procesan dichas aplicaciones (en particular, contenidos web). No se analiza aquí, por tanto, la influencia que la lógica remota del servicio pudiera tener en la accesibilidad en cuanto a la secuencia de interacciones del escenario de uso o a la generación dinámica de contenidos.

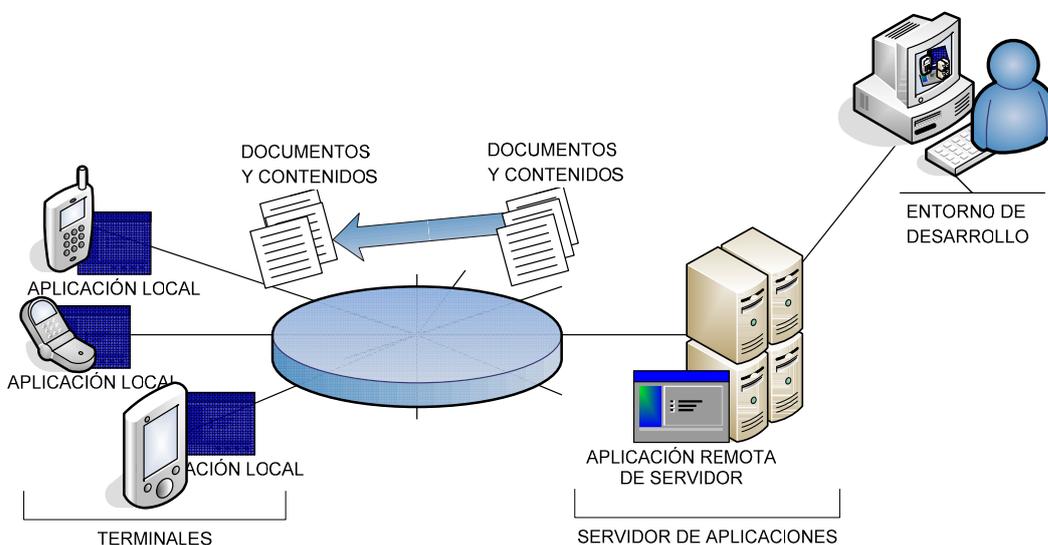


Figura 1. Marcos de despliegue y desarrollo de servicios para terminales móviles.

Dentro de este ámbito de aplicación de la herramienta (aplicaciones locales de terminal y contenidos), la existencia y madurez de ayudas técnicas limita las tecnologías del software objeto de evaluación por parte de la herramienta: aunque la accesibilidad de un servicio se debe evaluar *a priori* de forma agnóstica respecto a cualquier ayuda técnica concreta de que pueda disponer un usuario, no tiene sentido definir y determinar la accesibilidad de un servicio creado para plataformas que, de por sí, no son accesibles por no soportar ayudas técnicas. En otras palabras, los criterios de evaluación de la accesibilidad deben responder a la posibilidad real y verificable de que un servicio que los cumpla sea accesible, lo cual nunca es posible si la plataforma sobre la que se ejecuta no lo es. Por ello, las herramientas de evaluación de accesibilidad se deben dirigir a validar servicios para plataformas para las que se dispone de ayudas técnicas. Es notable que hoy en día sólo existen ayudas técnicas maduras para entornos Symbian: lectores de pantalla como Nuance TALKS [5] y Mobile Speak [6], o ampliadores software de pantalla como Mobile Magnifier [7]. Este aspecto resulta determinante en la definición de la herramienta y, por ello, el ámbito de los servicios objeto de evaluación de la herramienta estará restringido a entornos Symbian.

3. Herramientas existentes para la evaluación de accesibilidad

Para garantizar la accesibilidad a un servicio telemático con independencia del contexto de uso y de las capacidades del usuario, es necesario que la plataforma de acceso al servicio disponga de las pertinentes *ayudas técnicas* que se requieran y que el propio servicio haya sido diseñado cumpliendo con *criterios de diseño universal*, permitiendo que dichas ayudas técnicas adapten la presentación de la información al usuario, para que así éste pueda interactuar con el servicio. A pesar de que la accesibilidad –como categoría de requisitos que es– no admite una especificación única, sino que debe adaptarse a cada proyecto concreto; existen criterios comúnmente aceptados en la industria que especifican las características que hacen que un servicio sea accesible, así como herramientas software que apoyan la decisión sobre el cumplimiento de estos criterios, llevando a cabo una evaluación automática o semiautomática al respecto. Como se ha indicado, apenas existen herramientas de evaluación de accesibilidad de servicios para terminales móviles, lo que ha motivado el desarrollo de este trabajo. No obstante, se puede partir de pautas y herramientas existentes, destinadas a otras tecnologías afines, para extrapolarlas al ámbito del proyecto.

3.1. Pautas para la evaluación de accesibilidad.

En cuanto a los criterios o pautas de accesibilidad, existen pautas genéricas aplicables a todo desarrollo de ingeniería software y otras más específicas

relativas a tecnologías o productos concretos. En relación a la herramienta desarrollada, las pautas relevantes serán aquellas relacionadas con aplicaciones software y con contenidos de usuario (especialmente contenidos web). A continuación se presentan algunas de las familias más notables, empleadas para determinar las pruebas de accesibilidad que implementa la herramienta desarrollada.

La Web Accessibility Initiative (WAI) del World Wide Web Consortium (W3C) definió las **Web Content Accessibility Guidelines (WCAG) 1.0** [8], pautas de accesibilidad cuya aplicación se dirige a contenidos web. Las pautas son agnósticas respecto a la tecnología de representación de contenidos, aunque existen varios conjuntos de técnicas complementarias para lograr su cumplimiento en contenidos que emplean tecnologías concretas (HTML, hojas de estilo CSS – Cascading Style Sheets, lenguajes de *scripting*, etc.). La propia WAI-W3C ha definido también las **User Agent Accessibility Guidelines (UAAG) 1.0** [9], relativas a la accesibilidad de agentes de usuario de acceso a la web: definen tanto los requisitos de accesibilidad del propio agente de usuario, como los requisitos para que éste facilite la accesibilidad de los contenidos que presente.

La legislación estadounidense obliga al cumplimiento de requisitos de accesibilidad por parte de la Administración, y el correspondiente desarrollo reglamentario y normativo se recoge en lo que se conoce con el nombre de **Section 508** [10]. Éste incluye, entre otras, familias de pautas sobre aplicaciones informáticas y sistemas operativos, así como sobre sistemas de información y aplicaciones web (coincidente en muchos puntos con las WCAG 1.0). La Administración estadounidense mantiene un sitio web con formación e información respecto a la implementación de técnicas concretas para su cumplimiento.

Por último, las **Irish National Disability Authority (NDA) IT Accessibility Guidelines** [11] incluyen, entre otras, familias de pautas de accesibilidad referentes a tecnologías web (coincidente con las WCAG 1.0) y a software de aplicaciones.

Aparte se encuentran las **Mobile Web Best Practices (MWBP) 1.0** [12] de la Mobile Web Initiative (MWI) del W3C: se trata de una guía de recomendaciones para la mejora de la experiencia de usuario de la web móvil que, aunque no es específica sobre accesibilidad, sí incluye recomendaciones aplicables a esa categoría de requisitos y que, sobre todo, pueden servir de directrices respecto a la traslación de requisitos desde otros ámbitos al entorno móvil.

3.2. Aplicaciones software para la evaluación de accesibilidad.

Para evitar la tediosa tarea de verificar manualmente el cumplimiento de criterios de accesibilidad, existen herramientas software de evaluación de accesibilidad que, partiendo de un contenido,

determinan automáticamente si éste cumple con una serie de criterios programados en la herramienta. No obstante, no se puede evaluar cualquier criterio de accesibilidad de forma automática, con lo que la función de las herramientas de evaluación es realmente asistiva: determinan el cumplimiento de una parte de los criterios de accesibilidad y avisan al desarrollador de los lugares donde debe verificar otros criterios de forma manual.

Para el desarrollo de la herramienta de evaluación de accesibilidad para aplicaciones móviles presentada, nos hemos inspirado en herramientas de evaluación de accesibilidad existentes para otras tecnologías. Se han considerado especialmente las herramientas de evaluación de contenidos web: se han estudiado las características de aplicaciones de evaluación como TAW (Test de Accesibilidad Web) [13], HERA [14], Watchfire Bobby [15], Cynthia Says [16], y WAVE [17]. Todas ellas evalúan el cumplimiento de las pautas WCAG 1.0, y la mayoría también evalúa el cumplimiento de las pautas de la Section 508. Otro grupo de herramientas cuyo uso resulta relevante en el análisis de la accesibilidad web son los validadores [18], que no analizan la accesibilidad, sino que se encargan de validar la conformidad de los contenidos respecto a la especificación de los lenguajes en los que están escritos. Por último, se han tenido en cuenta *plug-ins* para entornos de desarrollo, dirigidos a integrar la evaluación automática de accesibilidad de aplicaciones software en las herramientas habituales de desarrollo: entre estos, destacan RAVEN (Ruled Based Accessibility Validation Environment) [19] para Eclipse y a11y [20] para Netbeans; ambos creados para evaluar la accesibilidad de las interfaces gráficas de usuario de aplicaciones Java de escritorio.

Los resultados de una evaluación de accesibilidad automatizada se suelen mostrar en el formato de un informe legible por un humano y que se pueda almacenar de forma persistente. Sin embargo, esto no sirve para procesar de forma automática el resultado de un informe, ni para intercambiarlo en distintas aplicaciones; para ello es necesario un lenguaje formal de representación de informes de accesibilidad. Este papel lo cumple el Lenguaje de Evaluación e Informes (Evaluation And Report Language – EARL) [21]. Se trata de un lenguaje definido por la WAI del W3C para la representación, en un formato procesable por una máquina, de los informes con los resultados de la evaluación del cumplimiento de criterios de accesibilidad o de otro tipo. EARL es una aplicación o *vocabulario* de Resource Description Framework (RDF), un modelo de metadatos basado en aserciones. Así, un informe EARL es un conjunto de aserciones que afirman el cumplimiento (o no) por un recurso dado de un conjunto determinado de criterios de accesibilidad. Cada vez más herramientas de evaluación de accesibilidad permiten la opción de almacenar en formato EARL (aunque con extensiones frecuentemente

propietarias) el contenido de un informe de resultados de evaluación de accesibilidad.

4. Definición de la herramienta de evaluación de accesibilidad para servicios para terminales móviles

4.1. Escenario de uso y roles en los procesos de negocio

El resultado del trabajo que se presenta ha consistido en crear una herramienta encargada de evaluar la accesibilidad de distintos componentes que intervienen en un servicio para terminales móviles. Aunque la funcionalidad última proporcionada a los usuarios de la herramienta consiste en ofrecerles información sobre la accesibilidad de un componente, los usuarios pueden desempeñar distintos roles dentro del proceso de desarrollo del servicio objeto de evaluación, lo que dará lugar a distintos escenarios de uso de la herramienta.

En el caso más general, aquí presentado, intervendrán múltiples agentes de la cadena de valor de prestación de servicios. Partiremos de un operador de telecomunicaciones móviles, que actúa como *proveedor de servicios*, y que desea que todos los servicios ofrecidos a sus usuarios sean accesibles, independientemente de que su desarrollo haya sido interno o bien haya sido externalizado a un tercero. Un *analista de requisitos* del operador determina que los servicios deben cumplir con un conjunto de criterios de accesibilidad ampliamente reconocido en la industria y además, desea añadir a ellos un conjunto propio de criterios de evaluación de accesibilidad, definidos en el ámbito corporativo, para ser aplicados específicamente a sus servicios. Para ello, un *desarrollador de pautas* del operador se encargará de implementar la evaluación de nuevas pautas extendiendo las interfaces que la herramienta proporciona al efecto. El operador proporciona la herramienta a los *desarrolladores de servicios* (internos o externos), junto con una especificación de las pautas de accesibilidad que deben cumplir los desarrollos y, en su caso, los análisis de pautas adicionales implementados por el desarrollador de pautas. Así, la herramienta –junto con las pautas de accesibilidad– facilita a los desarrolladores de servicios un contrato de especificación y validación de requisitos de accesibilidad que deben cumplir. Durante el desarrollo, un desarrollador de servicios emplea la herramienta de evaluación sobre los componentes del servicio que va creando para evaluar si cumplen el conjunto de pautas determinado. A continuación corrige los problemas de accesibilidad detectados en su caso por la herramienta. Al finalizar el desarrollo, si vuelve a emplear la herramienta, ésta deberá informar positivamente sobre la accesibilidad del servicio. A la entrega del servicio desarrollado, un *evaluador de accesibilidad* emplea de nuevo la herramienta para evaluar todos los servicios antes de su lanzamiento al mercado: en caso de que la herramienta proporcione una valoración positiva

(respecto a los criterios que determinó el analista de requisitos), el evaluador aprueba el servicio para que continúe hacia la puesta en producción y despliegue; en caso de que la herramienta detecte fallos, el evaluador remite al desarrollador del servicio un informe con los problemas detectados (a partir del propio informe emitido por la herramienta) y retorna a éste el flujo de trabajo para que corrija los errores sobre el servicio. En los desarrollos externalizados, pueden aparecer tanto evaluadores internos como externos. Una vez puesto en producción, durante la existencia en oferta del servicio, los **desarrolladores de mantenimiento** del servicio vuelven a emplear la herramienta tras cada cambio efectuado sobre éste, para asegurar la conservación de la accesibilidad de forma continuada; análogamente, el evaluador de accesibilidad puede volver a ejecutar el análisis tras cada cambio. En el caso de partir de sistemas heredados, son estos desarrolladores de mantenimiento los encargados de reconvertir los servicios existentes para lograr su accesibilidad, utilizando la herramienta de forma análoga a como hacen los desarrolladores de servicios en los nuevos servicios. En casos más sencillos que el presentado en este escenario, algunos de los roles pueden no existir, o bien pueden coincidir en el mismo agente.

4.2. Especificación de la herramienta

El escenario planteado en el epígrafe anterior, con sus posibles extensiones, presenta un variado conjunto de roles en relación con la motivación que cada uno de ellos tiene para el uso de la herramienta. Sin embargo, desde el punto de vista de su manejo, todos se reducen a un único papel: el del evaluador de accesibilidad (ya que el único rol que no encaja ahí, el desarrollador de pautas, no interactúa con la herramienta sino que la extiende). Así, según se presenta en la Fig. 2, el sistema interactúa con un único actor principal (evaluador), más un actor secundario (sistema de almacenamiento de recursos). El **evaluador** es un usuario humano – generalmente, un desarrollador de servicios móviles – encargado de dar la orden de ejecución de los procesos que ofrece la herramienta para efectuar un análisis de accesibilidad y/o presentar los resultados, así como de configurarla. El **sistema de almacenamiento de recursos** representa a la entidad en la que están almacenados los recursos que serán

objeto de evaluación y a la cual deberá acceder la herramienta para recuperarlos. Este sistema de almacenamiento de recursos abstrae del sistema real que se esté empleando, que puede ser tanto un sistema de ficheros local o remoto, como la World Wide Web, u otro sistema cualquiera.

La especificación de la herramienta se ha definido a partir de tres casos de uso: evaluación, presentación y modificación. Cada caso de uso constituye una secuencia con utilidad final para el usuario-evaluador, aunque están estrechamente relacionados entre sí y pueden resultar dependientes, según se ha mostrado en la Fig. 2.

El **caso de uso de evaluación** se dispara cuando el evaluador envía a la herramienta la orden de llevar a cabo una nueva evaluación. A continuación, el evaluador indica a la herramienta el ámbito de la evaluación (es decir, el conjunto de recursos objeto de evaluación) y los criterios de evaluación (es decir, el conjunto de pautas cuya validación se efectuará). Para ello, la herramienta proporciona al evaluador una serie de utilidades para definir el ámbito de evaluación según distintas restricciones (pertenencia a sitios o directorios, seguimiento de enlaces, tipos de recurso) y para seleccionar las familias de criterios de evaluación. Según se ha indicado, las tecnologías asistivas existentes (ayudas técnicas) imponen ciertos requisitos de plataforma en relación a los elementos objeto de análisis de la herramienta. Así, el prototipo está orientado a contenidos y aplicaciones para terminales que ejecuten la plataforma S60 de Nokia sobre el sistema operativo Symbian OS. La herramienta permite filtrar en la selección ficheros de tipo WML, XHTML, ficheros de descripción textual de recursos de GUI de Symbian y código fuente C++; utilizando para ello el tipo MIME y la extensión del fichero en su caso. Una vez que el evaluador ha configurado las características de la evaluación, ordena la ejecución del proceso. Entonces, la herramienta va recuperando de los sistemas de almacenamiento de recursos todos los recursos que se encuentran dentro del ámbito definido por el usuario, y va ejecutando la evaluación del cumplimiento de los criterios seleccionados. Cuando no queda ningún recurso por recuperar y evaluar, la aplicación almacena de forma persistente los resultados de la evaluación y el proceso finaliza.

El **caso de uso de presentación** se dispara cuando el evaluador envía a la herramienta la orden de presentar el resultado de una evaluación previa, cuya existencia resulta un prerequisite para este caso de uso. Dicha evaluación puede ser la que se acabe de efectuar o bien el resultado de una evaluación anterior –en ese caso, el evaluador indicará a la aplicación la ubicación de donde recuperar el resultado de dicha evaluación–. El evaluador indicará el tipo de presentación deseada y la aplicación presentará al usuario los resultados en el formato elegido. Dependiendo de características de la evaluación (tipo de recursos evaluados), existirán distintas posibilidades de presentación: siempre se

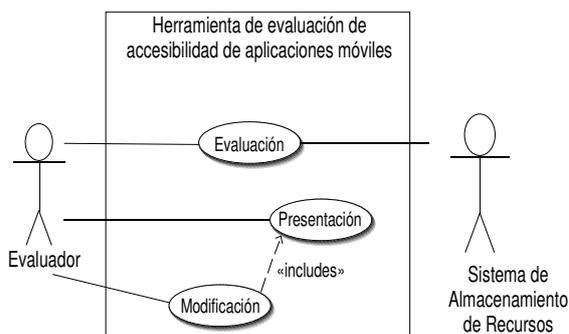


Figura 2. Diagrama de casos de uso de la herramienta de evaluación de accesibilidad de aplicaciones móviles.

podrá presentar un informe EARL en bruto, así como una vista del informe en forma de árbol navegable; además, en algunos contenidos el evaluador puede recibir también una vista de la GUI original en donde aparecen resaltados los elementos problemáticos.

El *caso de uso de modificación* incluye al de presentación como prerequisite. El evaluador, una vez que haya solicitado y recibido la presentación en forma de EARL o de árbol, puede modificar los contenidos de las aserciones generadas en la evaluación. Esto puede deberse a que el evaluador no desea efectuar la validación de algunos puntos, no está satisfecho con los criterios de evaluación empleados por la herramienta, o desea indicar el resultado de una comprobación manual en los puntos en donde sea necesaria. En cualquiera de los casos, la herramienta almacenará la aserción modificada, junto con la indicación de que ha sido generada manualmente.

Adicionalmente, se han considerado requisitos no funcionales de usabilidad de la propia herramienta de forma transversal a todos los casos de uso. Una característica adicional que se pretende implementar es la internacionalización del software, desacoplando la información localizada respecto del código fuente.

5. Arquitectura y diseño de la herramienta

El desarrollo de la herramienta se ha llevado a cabo siguiendo un diseño iterativo, y definiendo la arquitectura a partir de los casos de uso mencionados. Así, se ha dividido la herramienta, según un modelo estructurado, en un módulo encargado del proceso de ejecución y un módulo encargado de los procesos de presentación y modificación. El diagrama de flujo de datos resultante es el que se muestra en la Fig. 3, donde se puede observar que el resultado del proceso de evaluación se guarda en un almacén intermedio (“informe”) que sirve como entrada al proceso de presentación y cuyo contenido puede ser asimismo modificado desde el proceso de modificación.

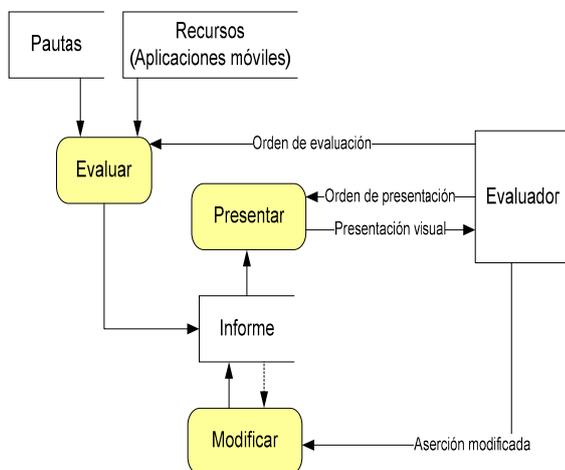


Figura 3. Diagrama de flujo de datos de primer nivel (DFD1) de la herramienta de evaluación de accesibilidad de aplicaciones móviles.

Cuando la aplicación ejecuta el proceso de evaluación, en primer lugar recupera los recursos objeto y a continuación aplica cada una de las pruebas definidas en las familias de pautas seleccionadas. La recuperación de los recursos se efectúa mediante una cola en la que se van introduciendo todos aquellos que encajen en el ámbito; el análisis de un recurso puede implicar añadir otros a la cola (por ejemplo, por el seguimiento de enlaces). La definición de las pruebas sigue la estructura definida en el siguiente pseudocódigo:

```

foreach element in scope()
  if condition(element)
  then return validityLevelA
  else return validityLevelB
  
```

Es decir, se selecciona un ámbito de aplicación de la prueba (por ejemplo, imágenes incrustadas) y, para cada elemento del ámbito, se comprueba la veracidad de una condición sobre él (por ejemplo, existencia de texto alternativo): en función del valor de dicha condición, se asigna uno u otro nivel de validez al resultado de la prueba. Los distintos resultados corresponden a los definidos en EARL: además de resultados de aceptación y fallo, existen resultados de indecisión, que se emplearán en las pruebas que requieran posterior verificación manual. En el caso de ficheros XML (WML y XHTML), tanto la selección del ámbito como la condición se expresan generalmente utilizando XPath como lenguaje de selección de elementos y de evaluación lógica. Esto permite tener un modelo de ejecución de pruebas reutilizable para distintas pruebas sin más que instanciarlo con distintos valores de las expresiones XPath [22]. En el caso de código C++, también existe una interfaz común de todas las pruebas, sin embargo, es necesario implementar cada una por separado. Para ello, se emplean bibliotecas de análisis de código en C++, que permiten acceder a la estructura y el valor de distintos elementos de un programa C++ de forma dinámica, sin conocer previamente su estructura. En concreto, se emplea la biblioteca PUMA (PURE manipulator), que proporciona un análisis semántico del código fuente y facilita un *framework* para realizar operaciones de manipulación sobre éste. En cuanto a los ficheros de descripción de recursos de Symbian, el análisis se efectúa a partir del resultado ofrecido por un compilador de dichos ficheros a una estructura de objetos Java, es decir, se representa la estructura de recursos como objetos Java estáticos. Dicho compilador se basa en un análisis léxico-sintáctico efectuado empleando JavaCC y JTree [23], a la que se añade un análisis semántico programado en Java de modo que implementen las reglas de definición de los ficheros de descripción de recursos Symbian.

Es frecuente (tanto en XML como en aplicaciones Symbian), que la aplicación de las pautas implique evaluar características tanto de un elemento como de otros con los que éste está relacionado (por ejemplo, de elementos situados antes o después del actual).

En ciertas ocasiones, la estructura común diseñada para las pruebas no es suficiente para determinar su resultado: es ese caso, se han diseñado clases específicas que implementan la prueba. Las pruebas efectuadas se han agrupado en distintos niveles: casos de prueba, técnicas y puntos de verificación, reflejando la estructura que las familias de pautas emplean habitualmente en su definición. Los resultados de las pruebas se representan siempre usando la misma estructura –independientemente de las pautas, el tipo de prueba y el recurso objeto de la evaluación–, la cual se representa en el diagrama de clases UML de la Fig. 4. La estructura de datos refleja la semántica definida por el lenguaje EARL de representación de resultados de evaluación de accesibilidad. Así, un informe de evaluación es un conjunto de aserciones con información adicional (fecha, evaluador, entorno, etc.): cada una de ellas incluye referencias al autor de la aserción (Assertor), a la prueba efectuada (TestCase) –indicando la familia de pautas y el test concreto aplicado– y al objeto de evaluación –indicando el recurso (Resource) y una referencia a una localización interna a éste (ResourcePointer)–, más los valores del nivel de fiabilidad (ConfidenceLevel), del resultado de la prueba (ValidityLevel) y la indicación de si es manual

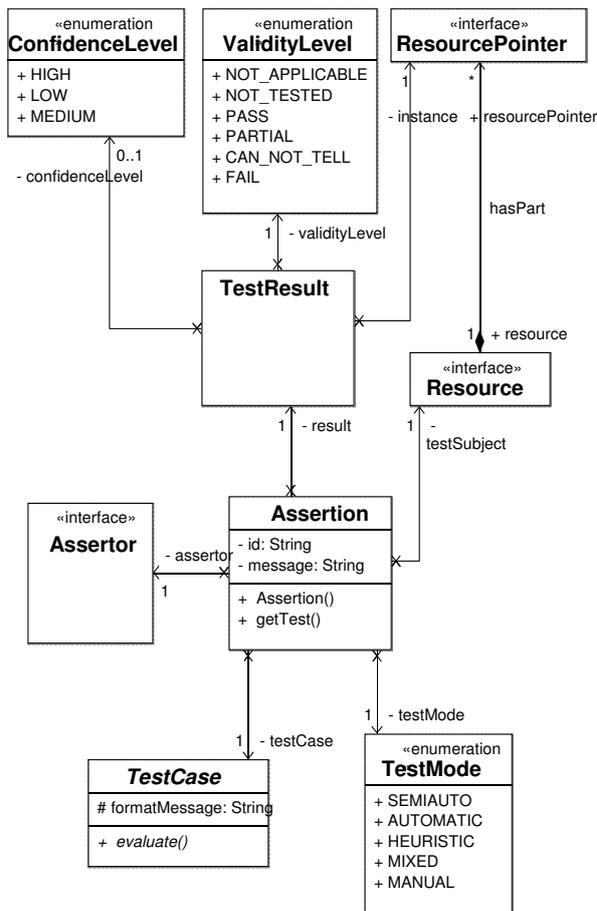


Figura 4. Diagrama de clases: asociaciones de la clase Assertion. Se presentan todos los elementos que forman parte de una aserción generada por la herramienta.

o automática (TestMode). Las referencias internas de un recurso se representan mediante XPointer [24] para el caso de XML, y mediante la posición y línea para el resto (código C++, descripción de recursos). El resultado de la prueba (“nivel de validez”) puede tomar distintos valores, según se ha indicado en la Fig. 4. Uno de esos valores (“CAN_NOT_TELL”) sirve para indicar una prueba cuyo resultado no se puede determinar completamente de forma automática. En ese caso, el proceso de presentación de la herramienta indicará al evaluador la necesidad de revisar manualmente el cumplimiento de un criterio dado, facilitando la referencia a la localización dentro del recurso donde se debe efectuar la revisión manual. Durante el proceso de modificación, el usuario puede introducir manualmente nuevas aserciones dentro del informe (cuyo TestMode quedará marcado con el valor “MANUAL”) para reflejar el resultado de las revisiones que haya efectuado manualmente.

En cuanto a la interfaz de usuario, se ha diseñado a partir de la plataforma Rich Client Platform (RCP) [25] de desarrollo de aplicaciones de Eclipse. Esto ofrece dos ventajas: el diseño con RCP permite reutilizar en la aplicación todos los *plug-ins* existentes para Eclipse, incluyendo componentes complejos de interfaz de usuario (UI) y APIs relacionadas (por ejemplo: gestión de preferencias, ayuda, editores XML); además, las herramientas RCP facilitan la integración futura de la herramienta como *plug-in* de un entorno de desarrollo basado en Eclipse. La interfaz tiene dos elementos básicos: la configuración de un análisis a partir de diálogos de propiedades –donde se eligen el ámbito y los criterios– y la presentación de los resultados del análisis en forma de árbol o informe EARL en bruto (se prevé también presentarlo sobre la propia UI evaluada).

Para implementar las pruebas se ha partido de familias de pautas ya existentes, filtrándolas, trasladándolas y adaptándolas a los tipos de contenidos evaluables. En el caso de WML, se ha partido de las pautas WCAG 1.0, de sus correspondientes técnicas nucleares y para HTML, y de la Section 508. Se han seleccionado los puntos de verificación y técnicas relevantes en WML y, apoyándose en una matriz de trazabilidad en relación a los elementos a los que cada uno pudiera afectar, y se han diseñado uno o varios *tests* para cada posible combinación. En el caso de XHTML-MP, se puede partir de *tests* ya existentes, por lo que sólo es necesario eliminar aquellos donde aparezcan elementos que existen en XHTML pero no en el perfil XHTML-MP. En cuanto a las aplicaciones para Symbian, no existe una familia concreta de pautas que encaje exactamente en las necesidades de su evaluación, por lo que las pruebas que se han creado han ido recogiendo y adaptando pautas procedentes de distintas familias relacionadas con aplicaciones software. En concreto, se ha partido de las pautas correspondientes de la Section 508, de la NDA y de las pautas UAAG 1.0 (que, si bien están

orientadas a navegadores, son parcialmente aplicables). En algunas de las pruebas para aplicaciones Symbian se parte de ficheros textuales que siguen un modelo declarativo y contienen descriptores de los recursos de UI empleados por una aplicación; mientras que en otras se parte del código fuente. En cualquier caso, no existe una distinción a priori respecto a qué pruebas para aplicaciones Symbian se efectúan sobre cada uno de los tipos de código, más allá de la facilidad de implementación de la prueba.

Adicionalmente a las pruebas basadas en las pautas que definen las distintas familias, se han ido definiendo también pruebas basadas en “pautas propias” de la herramienta. Estas pautas propias se han definido a partir de artículos sobre usabilidad en terminales móviles o a partir de la experiencia propia del uso de emuladores de distintos terminales, detectando aquellos puntos que podían dar problemas de accesibilidad en los terminales móviles pero que no habían sido tratados en las pautas para otras tecnologías que se han usado como referencia.

En cuanto al contenido concreto de las pautas, se pueden resumir los aspectos más relevantes que se han ido tratando. Las principales dificultades que pueden aparecer en cuando a la accesibilidad de WML detectadas tienen como origen las siguientes causas:

- Escasez de recursos de estilo, que limita las opciones posibles de presentación.
- Falta de separación entre estilo y contenido (no existen hojas de estilo).
- Contenido tempo-dependiente (navegación activada por temporizadores y eventos).
- Estructura de navegación intra-página (modelo de navegación por cartas).
- Dependencia respecto del dispositivo en los elementos soportados y su presentación.

En cuanto a XHTML-MP, al ser un subconjunto de XHTML, los problemas de accesibilidad serán análogos. Además, en el diseño de XHTML (y por tanto de XHTML-MP) ya se había tenido especialmente en cuenta la accesibilidad, por lo que los problemas no vienen tanto del lenguaje, sino del posible mal uso de sus elementos por parte del desarrollador.

6. Conclusiones y trabajos futuros

En esta ponencia se ha presentado un prototipo de herramienta de evaluación de accesibilidad de aplicaciones y contenidos para terminales móviles. La herramienta se concibió tras plantear una metodología de evaluación de accesibilidad y usabilidad de servicios y aplicaciones para terminales móviles, al determinar que el punto más relevante del proceso es la evaluación del cumplimiento de requisitos de accesibilidad. Siguiendo esa metodología definida, se han identificado los diferentes roles relacionados con la consecución de la accesibilidad que interactúan con

la herramienta y dónde intervienen en un proceso de negocio de desarrollo de servicios, determinándose los pasos que se deben introducir en el flujo de trabajo al emplear la herramienta.

Para el desarrollo de la herramienta, se ha llevado a cabo previamente un estudio de otras herramientas de evaluación de accesibilidad existentes en el mercado y dirigidas a distintas tecnologías. Asimismo, se han estudiado diversas familias de pautas de accesibilidad, se han seleccionado aquellas que podrían ser relevantes para los componentes de servicios móviles objeto de estudio en este trabajo y se han definido nuevas pautas resultantes de la traslación de las existentes en otras tecnologías al campo de los servicios móviles. La herramienta desarrollada implementa la evaluación automática de un subconjunto de las pautas seleccionadas y definidas en este trabajo; igualmente, presenta sugerencias respecto a los puntos que requieren de una intervención manual del evaluador para su comprobación.

Durante el desarrollo de la herramienta, se ha intentado mimetizar el comportamiento de otras herramientas de evaluación existentes para contenidos web y aplicaciones de PC. Sin embargo, esto no es posible de forma directa, ya que la plataforma de ejecución y la de evaluación en nuestro caso son distintas, lo que dificulta, por ejemplo, presentar los problemas de accesibilidad sobre una vista de la presentación original de los contenidos. Esto también introduce problemas en cuanto al desarrollo, por ejemplo, en cuanto a la manera de enlazar bibliotecas de manipulación de código para PC con bibliotecas para Symbian. Asimismo, se ha comprobado que las pautas de accesibilidad para otras tecnologías no son directamente trasladables a contenidos para móviles y, en muchas ocasiones, nos hemos encontrado con el problema de la complejidad que supone adaptar los criterios de comprobación de una pauta a una tecnología para la que no ha sido diseñada. Es especialmente relevante que las pautas WCAG 1.0, que se supone que deben ser válidas para todo tipo de contenidos web, realmente están orientadas a HTML –quizás por su antigüedad (1998)–, lo que de hecho ha llevado al W3C a estar definiendo una nueva versión. El apoyo en las MWBP no ha sido suficiente para resolver completamente este problema, ya que éstas directrices no están vinculadas específicamente a la accesibilidad, por lo que muchas no eran relevantes para nuestros objetivos.

El funcionamiento del prototipo de la herramienta realizada se ha validado aplicándola a servicios y contenidos reales, así como a un banco de pruebas diseñado *ex profeso* para esta herramienta. Se prevé ir añadiendo nuevas funcionalidades y soporte para nuevas familias de pautas en el futuro próximo, así como la mejora de aspectos como la accesibilidad de la propia aplicación y la internacionalización. En particular, se pretende añadir soporte para otras tecnologías de definición de contenidos web y otros

sistemas operativos para los que empiezan a aparecer ayudas técnicas (como Windows Mobile); así como para otros canales (por ejemplo, para contenidos definidos en SMIL – Synchronized Multimedia Integration Language y mensajes multimedia MMS).

Agradecimientos

El trabajo presentado en esta ponencia ha sido desarrollado en el contexto del proyecto “Marco para la evaluación de la usabilidad y accesibilidad en aplicaciones y servicios móviles”, financiado por la Cátedra Amena en la Universidad Politécnica de Madrid (actualmente Cátedra Orange).

Referencias

- [1] COST 219ter. Accessibility for All to Services and Terminals for Next Generation Networks. Disponible en <http://www.tiresias.org/cost219ter/>.
- [2] *Wireless Markup Language. Version 2.0. [Wireless Application Protocol. WAP-238-WML-20010911-a]*. Wireless Application Protocol Forum, 2001. Disponible en <http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf>.
- [3] M. Baker, M. Ishikawa, S. Matsui, P. Stark, T. Wugofski, T. Yamakami (eds.), *XHTML™ Basic. W3C Recommendation*. World Wide Web Consortium, 2000. Disponible en <http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/>.
- [4] *XHTML Mobile Profile. [Wireless Application Protocol. WAP-277-XHTMLMP-20011029-a]*. Wireless Application Protocol Forum, 2001. Disponible en <http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf>.
- [5] Nuance TALKS™. Nuance Communications. Disponible en <http://www.nuance.com/talks/>.
- [6] MobileSpeak. Code Factory. Disponible en http://www.codefactory.es/es/mobile_speak/mspeak.htm.
- [7] Mobile Magnifier. Code Factory. Disponible en http://www.codefactory.es/es/mobile_magnifier/mmagnifier.htm.
- [8] W. Chisholm, G. Vanderheiden, I. Jacobs (eds.), *Web Content Accessibility Guidelines 1.0. W3C Recommendation*. World Wide Web Consortium, 1999. Disponible en <http://www.w3.org/TR/WAI-WEBCONTENT/>.
- [9] I. Jacobs, J. Gunderson, E. Hansen (eds.), *User Agent Accessibility Guidelines 1.0. W3C Recommendation*. World Wide Web Consortium, 2002. Disponible en <http://www.w3.org/TR/UAAG10/>.
- [10] Estados Unidos: *Section 508 of the Rehabilitation Act*, 1973, según redacción enmendada por la *Workforce Investment Act*, 1998 (P.L. 105-220). United States Code (U.S.C.), título 29, sección 794d, 1998. Disponible en <http://www.section508.gov/index.cfm?FuseAction=Content&ID=3>.
- [11] *Irish National Disability Authority IT Accessibility Guidelines, Version 1.1*. Irish National Disability Authority. Disponible en <http://accessit.nda.ie/>.
- [12] J. Rabin, C. McConville (eds.), *Mobile Web Best Practices 1.0. Basic Guidelines. W3C Candidate Recommendation* [trabajo en curso]. World Wide Web Consortium, 2006. Disponible en <http://www.w3.org/TR/mobile-bp/>.
- [13] Test de Accesibilidad a la Web (TAW). Fundación CTIC. Disponible en <http://www.tawdis.net/>.
- [14] Benavidez, C., *Revisando la Accesibilidad con Estilo. HERA 2.0*. Fundación Sidar, 2005. Disponible en <http://www.sidar.org/hera/>.
- [15] WebXACT. Watchfire Corporation, 2004. Disponible en <http://webxact.watchfire.com/>.
- [16] HiSoftware® Cynthia Says™ Portal. Disponible en <http://www.contentquality.com/>.
- [17] WAVE 3.0 Accessibility Tool. Web Accessibility in Mind (WebAIM) Disponible <http://www.wave.webaim.org/index.jsp>.
- [18] Markup Validation Service. World Wide Web Consortium (W3C). Disponible en <http://validator.w3.org/>.
- [19] IBM Rule-based Accessibility Validation Environment. [RAVEN]. IBM, 2006. Disponible en <http://www.alphaworks.ibm.com/tech/raven>.
- [20] Accessibility Developer's Corner. Netbeans.org, 2006. Disponible en <http://a11y.netbeans.org/>.
- [21] C. McConville, S. Abou-Zahra (eds.), *Evaluation and Report Language (EARL) 1.0 Schema. W3C Working Draft*. World Wide Web Consortium, 2006 [trabajo en curso]. Disponible en <http://www.w3.org/TR/EARL10/>.
- [22] J. Clark (ed.), *XML Path Language (XPath) Version 1.0. W3C Recommendation*. World Wide Web Consortium, 1999. Disponible en <http://www.w3.org/TR/xpath>.
- [23] Java Compiler Compiler [tm] (JavaCC [tm]) - The Java Parser Generator. CollabNet. Disponible en <https://javacc.dev.java.net/>.
- [24] P. Grosso, E. Maler, J. Marsh, N. Walsh (eds.), *XPointer Framework. W3C Recommendation*. World Wide Web Consortium, 2003. Disponible en <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>.
- [25] J. McAffer, J.M. Lemieux, *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java™ Applications*. Addison-Wesley Professional, 2005.