

**ELENA GARCÍA
J. RAMÓN HILERA**

DEPT. CC. DE LA COMPUTACIÓN
ESCUELA POLITÉCNICA
CAMPUS UNIVERSITARIO
UNIVERSIDAD DE ALCALÁ
28871 MADRID

{elena.garciab, jose.hilera}@uah.es

MIGUEL A. SICILIA

DEPT. INFORMÁTICA, AV. UNIVERSIDAD, 30
UNIVERSIDAD CARLOS III
28911 MADRID
msicilia@inf.uc3m.es



El WAI ha dispuesto un buen número de recomendaciones con el objeto común de garantizar la accesibilidad en los diferentes niveles de la arquitectura Web.

VERIFICACIÓN DE LA ACCESIBILIDAD A INTERNET

DEBIDO A LA DIVERSIDAD DE DISPOSITIVOS [6] (DESDE NAVEGADORES EN TELÉFONOS MÓVILES HASTA LECTORES DE PANTALLA PARA PERSONAS INVIDENTES), A LA RÁPIDA EVOLUCIÓN DE LAS TECNOLOGÍAS WEB [5] Y A LAS CARACTERÍSTICAS DE LOS DIFERENTES USUARIOS (POTENCIALES Y REALES) DE LA MISMAS, LA CLAVE DE LA CALIDAD DE LAS INTERFACES DE USUARIO DE UN SITIO O APLICACIÓN WEB EMPIEZA A REQUERIR EL USO DE OTRAS PRÁCTICAS COMPLEMENTARIAS A LA USABILIDAD COMO ES LA ACCESIBILIDAD [8], ENTENDIDA COMO LA CAPACIDAD DE LAS APLICACIONES PARA SER UTILIZADAS POR UN AMPLIO RANGO DE USUARIOS EN UN AMPLIO ESPECTRO DE SITUACIONES.

ENTORNOS DE DESARROLLO WEB Y ACCESIBILIDAD

El World Wide Web Consortium (W3C) promueve la Iniciativa para la Accesibilidad en Web (WAI) como un foro de discusión y divulgación de las prácticas de accesibilidad. El WAI ha dispuesto un buen número de recomendaciones con el objeto común de garantizar la accesibilidad en los diferentes niveles de la arquitectura Web. Actualmente, las más importantes son las que versan sobre agentes de usuario [12] (propuesta de

recomendación), sobre contenidos en formato Web [13] y sobre las herramientas de autor [10]. Dicho organismo también mantiene documentos públicos sobre las características de accesibilidad del lenguaje de sincronizado multimedia (SMIL) [16] y sobre las características de accesibilidad del formato SVG [17].

Muchas de las aplicaciones para Internet o Intranet, ya sea desarrolladas para uso interno de la empresa o para la publicación de contenidos, se construyen con la ayuda de un entorno de desarrollo Web (EDW). Ejemplos de estos entornos son Microsoft Visual InterDev, Allaire ColdFusion, IBM Websphere o

DreamWeaver Ultradev de Macromedia. Un EDW habitualmente combina editores HTML/XML/CSS propios de las herramientas de autor con:

- Soporte para programación de servidor, tales como servlets de Java, scripts de servidor (jsp, asp) o componentes COM combinados con páginas Web.
- Soporte para prácticas de ingeniería del software, como gestión de versiones, gestión de configuración o herramientas de depuración.

Llamaremos aplicación Web (AW) al producto de un EDW. Estas aplicaciones contienen tanto contenido HTML estáti-



co como código de servidor, y, por lo general, tienen usos y propósitos que difieren de los tradicionales sitios Web.

TRABAJO RELACIONADO

Siguiendo análisis previos [2, 3], consideramos que la adaptación de los EDW es un factor clave para garantizar la accesibilidad, ya que sus productos son las aplicaciones que están presentes en muchas de las organizaciones que se valen de la informática para su funcionamiento.

Existen en el mercado diferentes herramientas que permiten verificar la accesibilidad de las páginas Web. Estas herramientas pueden estar o no integradas en un entorno. Un ejemplo de herramienta de verificación externa es el servicio Bobby (www.cast.org/bobby), que comprueba la accesibilidad de ficheros HTML a partir de su URL. Existen herramientas que tienen integradas características de accesibilidad en el propio entorno, como son HotMetal Pro y el editor del W3C Amaya [7], pero ninguna de ellas se puede considerar un EDW, sino más bien herramientas de autor HTML. Otro trabajo muy significativo en el tema que se trata es el proyecto *A-Prompt* [1]. En este proyecto se desarrolla un conjunto de componentes binarios que verifican algunas pautas de accesibilidad en archivos (o fragmentos) HTML, lo que implica una adaptación de las versiones de los EDW por parte del

fabricante. Dentro de la línea de la verificación de la accesibilidad en EDW también hay que mencionar las extensiones que algunas compañías han construido para tal fin, como la generada por UsableNet para el entorno Macromedia DreamWeaver Ultradev1, que verifica los contenidos generados pero no permite una fácil adaptación a futuras directrices ni se basa en un formato para las recomendaciones que permita su reutilización en otros entornos de desarrollo. El objetivo final que se persigue en este trabajo consiste en adaptar los EDW para integrar la accesibilidad en el proceso de desarrollo de aplicaciones Web sin necesidad de rehacer las versiones de los mismos y de manera que se puedan manejar fácilmente futuras recomendaciones independientemente del entorno de desarrollo. El prototipo que hemos implementado tal efecto habilita, por el momento, la verificación y corrección de los contenidos estáticos de ficheros que potencialmente pueden contener código de servidor (jsp, asp,..).

ADAPTACIÓN A LA ACCESIBILIDAD DE LOS ENTORNOS DE DESARROLLO WEB

• ARQUITECTURA GENÉRICA DE UN ENTORNO DE DESARROLLO WEB.

Analizando las funciones comunes de los EDW comerciales, hemos identificado una serie de aspectos que han de ser contemplados en la generación de aplicaciones accesibles. La Figura 1 (de la página siguiente) muestra la arquitectura genérica de alto nivel de un entorno de desarrollo Web en la que se han incluido los elementos necesarios para su adaptación a la accesibilidad.

Un entorno de desarrollo Web está compuesto básicamente por un gestor de proyectos y varios editores integrados. El gestor de proyecto se conecta con un directorio del servidor Web y posibilita el desarrollo en grupo. Cuando un desarrollador solicita modificar cualquier componente de la aplicación web (una página, código,...), se realiza una

• Verificación de la Accesibilidad a Internet •

RESUMEN

Para generar aplicaciones accesibles se requiere el cumplimiento de algunas recomendaciones sobre los contenidos escritos por el desarrollador o generados por la herramienta de desarrollo Web. En este trabajo se presenta la descripción de un enfoque práctico y extensible para integrar las directrices de la Iniciativa para la Accesibilidad Web del W3C en entornos de desarrollo Web comerciales –que pueden utilizarse como soporte en un proceso de ingeniería del software– mediante la comprobación automatizada de contenidos basándose en una abstracción en formato XML de un subconjunto de las recomendaciones del W3C.

ABSTRACT

In order to generate accessible applications, some recommendations about contents, either written by developers or automatically written by the tool, must be fulfilled. In this work, we present the description of a pragmatic and extensible approach to integrate W3C Web Accessibility Initiative recommendations in commercial Web development environments, which can be used to support the software engineering process. Integration is carried out by means of automated content verification functionalities based on an XML information model of a subset of W3C recommendations.
KEYWORDS: Accessibility, Web applications, Web development environment, XML.

PALABRAS CLAVE

ACCESIBILIDAD,
APLICACIONES WEB
ENTORNOS DE DESARROLLO
WEB, XML

• Verificación de la Accesibilidad a Internet •

CONCLUSIONES

La primera conclusión que se obtiene de nuestro trabajo es que muchas de las directrices para garantizar la accesibilidad pueden ser automatizadas utilizando formatos extensibles persistentes. No obstante, hay algunas de ellas que continúan siendo responsabilidad del desarrollador. Por lo tanto, si el EDW proporciona los mecanismos de extensibilidad apropiados, construir un algoritmo para verificar la accesibilidad de código estático en aplicaciones web es relativamente sencillo.

La segunda de las conclusiones es que al utilizar los mecanismo de extensibilidad de los EDW, nuestro enfoque se desarrolla como una extensión adicional de las versiones actuales de los entornos, sin que sea necesario rediseñar el entorno ni sacar una nueva versión de la herramienta. Otro aspecto relevante de estos prototipos es que pueden ser extendidos para manejar cualquier nueva pauta que surja en el futuro a medida que avance la tecnología Web, ya que la ejecución de las reglas se lleva a cabo de forma genérica y semi-automatizada, independientemente de su contenido. De hecho, se pueden incorporar nuevos sujetos para reglas sin que su definición y posterior tratamiento sea excesivamente costoso. Un ejemplo que ilustra este aspecto sería la inclusión de un subconjunto parcial de las recomendaciones para obtener contenidos multimedia accesibles [16]. Sin hacer un análisis exhaustivo, algunas dichas recomendaciones pueden ser formalizadas en reglas con los formatos ya expuestos: <SUBJECT tag="VIDEO"

ARQUITECTURA GENÉRICA ADAPTADA DE EDW



(Figura 1)

copia local en su estación de trabajo y cuando se terminan las modificaciones pertinentes, esta copia se actualiza en el servidor Web. Los editores, que pueden estar poco o muy integrados en el EDW, trabajan sobre las copias locales.

Hemos utilizado el término genérico *capa de adaptación* para designar a todos los componentes que se deben añadir o modificar en el entorno con el fin de proporcionar o verificar la accesibilidad. Idealmente, esta capa de adaptación permite interactuar directamente con los contenidos mostrados en el editor, pero si el EDW carece de los mecanismos de extensibilidad necesarios para esta interacción, la capa de adaptación se ve forzada a operar directamente sobre las copias locales.

Los sistemas de ayuda son a menudo aplicaciones separadas que se invocan desde el entorno de desarrollo con alguna información de contexto (como por ejemplo la ayuda MSDN de Microsoft o los sistemas de ayuda en formato HTML como el que proporciona Macromedia). Por lo tanto, la adaptación de la ayuda puede realizarse de forma independiente.

• EL MOTOR DE EJECUCIÓN DE REGLAS DE CONTENIDO.

El componente principal de la capa de adaptación de un EDW para la verificación y corrección de la accesibilidad de los contenidos estáticos de una AW es el motor de ejecución de reglas de contenido (MRC). Este módulo opera sobre los contenidos de una aplicación Web de dos posibles formas:

- Verificando la accesibilidad de dichos contenidos una vez escritos y bajo petición del desarrollador.
- Realizando una verificación "intrusiva", es decir, verificando la accesibilidad de los contenidos a medida que se van escribiendo, mediante la captura determinados eventos como la inserción de una etiqueta de cierre u operaciones de *arrastrar y soltar*.

Cuando se detecta de cualquiera de las dos formas algún contenido no accesible, se ejecutarán las acciones correspondientes para realizar la corrección oportuna.

PRINCIPALES COMPONENTES DEL MRC

(Figura 2)



Componentes del Motor de Ejecución de Reglas de Contenido

“ El objetivo final que se persigue en este trabajo consiste en adaptar los EDW para integrar la accesibilidad en el proceso de desarrollo de aplicaciones Web”

El motor de ejecución de reglas de contenido (ver Figura 2) toma fragmentos de código como entrada, los examina y si son inaccesibles devuelve automáticamente una versión correcta de los mismos o dispara algún tipo de acción para corregirlos.

El módulo *verificador* contempla los documentos HTML/XML como un documento cuyo contenido se estructura en forma de árbol, tal y como lo exponen los analizadores *sintácticos para DOM* (Document Object Model [11]). Un fragmento de contenido se representa como un árbol jerárquico cuyos nodos son los elementos. Estos nodos se corresponden con las etiquetas del documento, y los atributos de cada etiqueta son representados como colecciones de pares (clave, valor) dentro del correspondiente nodo. La *base de datos de reglas* contiene una definición en XML [14] (o cualquier otro formato persistente) de las pautas de accesibilidad que se han de aplicar sobre estos árboles. De acuerdo con las reglas de la base de datos, el módulo verificador modifica directamente el árbol del documento o solicita algún tipo de información al desarrollador a través de cuadros de diálogo (activados gracias al *ejecutor de diálogos*). Esto diálogos se almacenan en un repositorio de componentes separado en formato binario o *“script”*.

El diseño de la base de reglas es el resultado de clasificar los puntos de verificación de las directrices del WAI, dependiendo de la parte estructural de HTML involucrada y el tipo de acción correctiva requerida. Una regla se aplica a un sujeto, es decir a un elemento HTML específico o una relación entre dos de ellos. Un análisis detallado de las Directrices para la Accesibilidad de Contenidos Web (WCAG) sugiere los siguientes sujetos de reglas:

- **Elemento simple.** Este es el sujeto indicado cuando la acción recae

sobre una única etiqueta. Por ejemplo, la etiqueta `<OBJECT>`, que requiere una descripción textual entre los delimitadores de la etiqueta.

- **Atributo en el contexto de un elemento.** Este sujeto indica que la acción se ha de aplicar a un atributo concreto de una etiqueta concreta. Por ejemplo, la etiqueta `` necesita un atributo `“alt”` no vacío que garantice su accesibilidad.

- **Pares de elementos.** Este sujeto indica que la acción se centra en la relación entre dos etiquetas. Por ejemplo, la presencia de una etiqueta `<SCRIPT>` requiere una etiqueta `<NOSCRIPT>` inmediatamente siguiente para asegurar que la página es accesible cuando los *scripts* no están soportados por el navegador.

- **Dependencia de atributos.** Este es el tipo de sujeto que indica que la regla se debe aplicar a etiquetas que manejan el atributo A2 sólo en función de otro atributo A1. Nótese que ambos atributos pertenecen a la misma etiqueta. Por ejemplo, en una etiqueta `<BUTTON>` se debe verificar la existencia de un atributo `“onkeypress”` solamente si existe el atributo `“onclick”`.

El tipo de acción correctiva que se ha de realizar cuando una regla se dispara puede ser una de las cuatro siguientes:

- **Asegurar.** Esta acción fuerza al desarrollador a incluir una etiqueta, atributo o contenido dentro de la etiqueta, de acuerdo con la estructura de la regla.
- **Prohibir.** Se utiliza para evitar la presencia de un atributo, etiqueta o relación estructural entre elementos.
- **Sugerir.** Esta acción invoca al ejecutor de diálogos para avisar al desarrollador sobre algún aspecto de accesibilidad recomendable pero no esencial.

```
attribute="abstract"> Att in element
</SUBJECT>
```

```
<CORRECTIVEACTION> Ensure
</CORRECTIVEACTION>
```

O bien se puede crear otro sujeto denominado elementos anidados que permita formalizar recursivamente aspectos como:

```
<SUBJECT tag="PAR" tag="
textstream"> Nested elements
</SUBJECT>
```

```
<CORRECTIVEACTION> Ensure
</CORRECTIVEACTION>
```

En cuanto a trabajo futuro, la mayor limitación reside en la verificación de código de servidor. Es inviable construir una herramienta que permita la verificación y corrección de cualquier tipo de componentes de servidor que generen código HTML, puesto que en última instancia éstos son desarrollados de acuerdo a estilos de programación personales que hacen imposible verificar su contenido por inspección. La solución parcial sobre la que se está trabajando se basa en facilitar las pruebas sistemáticas desde el entorno. Estas pruebas se llevarían a cabo mediante las bibliotecas HTTPUnit, orientadas a la filosofía de pruebas XP [18]. Actualmente estamos desarrollando un asistente que se ejecuta cuando comienza la verificación y que permite al desarrollador introducir, entre otros datos necesarios, el nombre y los parámetros de la página jsp para generar el esqueleto de una clase Java que constituye el caso de prueba. El caso se encargaría de ejecutar la página en el servidor y recoger su salida en formato HTML, con el fin de construir su árbol DOM. Dicho árbol se verificaría utilizando el MRC y el resto de componentes expuestos en este trabajo.

• Verificación de la Accesibilidad a Internet •

BIBLIOGRAFÍA

- A-Prompt Toolkit, disponible en <http://aprompt.snpw.toronto.ca/>
- García B., E., and Sicilia, M.A.: Análisis de los problemas de accesibilidad en Internet desde el punto de vista del empleo. En: Actas del Congreso Iberoamericano de Informática Educativa Especial, Córdoba, España (2000).
- García B., E., Sicilia, M.A., Hilera, J.R.: La Intranet como soporte a las aplicaciones de gestión en la empresa ¿una nueva barrera para las personas con discapacidad?. En: Actas de III Jornadas de Informática y Sociedad, Madrid, España (2000).
- Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison Wesley (1999).
- Nielsen, J.: Changes in Web Usability Since 1994. Jakob Nielsen's Alertbox (Diciembre 1997), disponible en <http://www.useit.com/alert-box/9712a.html>
- Nielsen, J.: The Difference Between Web Design and GUI Design. Jakob Nielsen's Alertbox (Mayo 1997), disponible en <http://www.useit.com/alert-box/9705a.html>.
- Quint, V., Vatton, I.: An Introduction to Amaya. W3C NOTE (1997).
- Nielsen, J. Disabled Accessibility: The Pragmatic Approach. Jakob Nielsen's Alertbox (Junio 1999), disponible en <http://www.useit.com/alert-box/990613.html>
- Pressman, R.: Software Engineering: A Practitioner's Approach. McGrawHill (1997).
- Treviranus, J. et al: Authoring Tool Accessibility Guidelines 1.0 (ATAG). W3C Recommendation. (2000).
- Wood, L. et al: Document Object Model (DOM) Level 1 Specification Version 1.0. W3C Recommendation (1998).
- Gunderson, J., Jacobs, I.: User Agent Accessibility Guidelines 1.0 (UAAG). W3C Proposed Recommendation (2000).
- Chisholm, W. et al: Web Content Accessibility Guidelines 1.0 (WCAG). W3C Recommendation (1999).

Algunos puntos de WCAG con prioridad 3 son modelados de esta forma.

- **Ejecutar algoritmo externo.** Esta acción ejecuta un programa externo para verificar determinados aspectos de la accesibilidad. Un ejemplo de programa externo es el algoritmo Gunning-Fog, que se utiliza para medir el grado de comprensión de un texto, indicando al desarrollador si es poco o muy comprensible. Como ejemplo, a continuación se muestra una regla correspondiente a parte del punto de verificación 1.1 de la directriz 1 del WCAG, que contempla la necesidad de un atributo "alt" dentro de las etiquetas IMG:


```
<RULE guideline="1" checkpoint="1.1.a" priority="1">
<SUBJECT tag="IMG" attribute="alt">
Att in element </SUBJECT>
<CORRECTIVEACTION> Ensure
</CORRECTIVEACTION> </RULE>
```

Basándonos en esta regla, podemos ilustrar el comportamiento del MRC (ver Figura 3) como sigue: Cuando un desarrollador solita verificar la presencia de textos alternativos se construye una consulta para recuperar las reglas que pertenezcan a la primera directriz de WCAG. Una de las reglas encontradas será la citada anteriormente, que especifica el sujeto atributo en el contexto de un elemento, donde el elemento es IMG y el atributo, "alt".

El siguiente paso consiste en buscar elementos IMG en el árbol DOM del

documento mostrado en la ventana actual. Si se encuentra algún elemento IMG en este árbol, se ha de recuperar la acción correctiva asociada a la regla. Como en nuestro caso la acción es asegurar, debe existir en el elemento un atributo "alt" no vacío. Si no fuera así, un cuadro de dialogo obliga al desarrollador a introducir un texto alternativo para la imagen.

El MRC actúa de una forma muy similar para procesar cualquier otro tipo de reglas, como el que se muestra a continuación, que impide que el atributo "target" del elemento <A> pueda adoptar el valor "_blank":

```
<RULE guideline="10" checkpoint="1" priority="2">
<SUBJECT tag="A" attribute="target">
Att in Element</SUBJECT>
<CORRECTIVEACTION value="_blank"> Avoid
</CORRECTIVEACTION> </RULE>
```

Como ya se ha expuesto, la adaptación del entorno conlleva también la adaptación de la ayuda de acuerdo a la 6ª pauta de las Directrices para la Accesibilidad de las Herramientas de Autor (ATAG). Para potenciar la accesibilidad en la ayuda y documentación, durante la instalación del prototipo se incluye en el menú de ayuda del entorno la opción "Desarrollo de Aplicaciones Accesibles", que muestra documentación en formato hipertextual sobre las técnicas y recomendaciones de accesibilidad del W3C en cuanto a contenidos

EJEMPLO DE COMPORTAMIENTO DEL MRC

(Figura 3)



Web y cómo éstas pueden ser verificadas sobre el código estático generado por la herramienta o por el desarrollador. De igual forma, la interfaz del prototipo incluye ayuda específica sobre los diferentes aspectos que pueden analizarse mediante él. Parte de dicha información se obtiene recuperando el número de directriz y de punto de verificación de las reglas correspondientes a las pautas que se verifican en las cada una de las opciones (atributos "guideline" y "checkpoint" de las etiquetas RULE). Los cuadros de diálogo que se activan desde la interfaz principal incorporan también información sobre el aspecto concreto al que hacen referencia, mostrándose la documentación de la misma manera que en la interfaz principal, con la única diferencia de que se obtiene el enlace a partir de la regla en curso y no desde un conjunto de reglas implicadas.

INTEGRACIÓN DE LA ACCESIBILIDAD EN EL CICLO DE VIDA DEL SOFTWARE

Puesto que, como se mencionó en un principio, consideramos la accesibilidad como un factor de calidad del software, los principios de garantía de calidad del software (SOA) deben tener en cuenta la accesibilidad en aplicaciones Web. Más concretamente, se debe tener en cuenta el clásico modelo de amplificación de errores descrito en [9]. Este modelo establece que no solamente los errores son más caros de corregir en etapas tardías del desarrollo sino que los errores al principio del ciclo de vida de una aplicación se propagan a etapas posteriores amplificándose en número.

Nuestra propuesta global integra la accesibilidad en el proceso de construcción del software tan pronto como sea posible. Hemos escogido como marco de referencia el Proceso Unificado [4], pero el modelo es directamente aplicable a cualquier otro método. Como se expone a continuación, tan sólo la fase de análisis es independiente de las consideraciones de la accesibilidad.

En la fase de requisitos, la accesibilidad se ha de considerar un requisito no fun-



(Figura 4)

Interfaz principal del prototipo para verificar la accesibilidad en Macromedia DreamWeaver UltraDev.

cional; en la fase de diseño las recomendaciones se utilizan para diseñar las interfaces de usuario; la fase de implementación esta guiada por un EDW adaptado para la accesibilidad y durante las pruebas se comprobarán características adicionales de accesibilidad.

Como resultado del proceso propuesto, la accesibilidad queda garantizada después de la fase de implementación gracias a la adaptación y automatización de la verificación en el entorno, y por lo tanto, se necesita muy poco esfuerzo para corregirla durante la fase de pruebas.

Una vez que el EDW está adaptado, el esfuerzo que debe realizar una organización de desarrollo de software para terminar de adaptar su proceso a la accesibilidad debería dirigirse en dos direcciones:

- Introducir la "cultura de la accesibilidad" y formar a los desarrolladores en las recomendaciones de WAI.
- Adaptar las normativas internas de estilo para las interfaces de usuario añadiendo las consideraciones del citado consorcio.

UN EJEMPLO DE ADAPTACIÓN DE EDW: MACROMEDIA DREAMWEAVER ULTRADEV

Nuestras ideas han sido probadas en el entorno Microsoft Visual InterDev [15] y más recientemente en Macromedia DreamWeaver UltraDev.

La interfaz principal del prototipo (ver Figura 4) es una ventana de paleta flotante.

• Pemberton, S. et al: XHTML(tm) 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0. W3C Recommendation (2000).

• García Barriocanal, E., Sicila Urbán, M. A., Hilera González, J. R.: Adapting the Web Development Environment for Accessible and Usable Application Construction. En: Actas de Webnet 2000 World Conference, San Antonio, Estados Unidos (2000).

• Koivunen, M.: Accessibility Features of SMIL. W3C Note (1999).

• McCathieNeville, C., Koivunen, M.: Accessibility Features of SVG. W3C Note (2000).

• Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley (2000)

Este es el tipo de interfaces que se utilizan en entornos Macromedia para mostrar las diferentes funcionalidades (inserción de objetos en una aplicación o modificación de los elementos ya existentes mediante sus propiedades, por ejemplo). Una paleta flotante es un fichero HTML que se puede asociar a determinadas opciones del menú del entorno y que contiene un formulario HTML y manejadores de eventos que se asocian a sus diferentes elementos.

Los usuarios de Macromedia Dreamweaver UltraDev pueden verificar la accesibilidad de los contenidos seleccionando todos o alguno de los diferentes temas de accesibilidad mostrados en la paleta. Cuando se pulsa en el botón "Comenzar la Verificación" se captura el árbol DOM del documento mostrado en la ventana actual del entorno. El algoritmo de verificación recupera entonces las reglas que pertenezcan a la directriz asociada a la opción seleccionada y el MRC las aplica secuencialmente sobre los nodos del árbol utilizando métodos de filtrado del propio esquema de DOM.

Como ya se ha descrito, la acción correctora se dispara de acuerdo con el sujeto de la regla y la estructura del nodo en estudio.