



RIED. Revista Iberoamericana de Educación a  
Distancia  
ISSN: 1138-2783  
ried@edu.uned.es  
Asociación Iberoamericana de Educación  
Superior a Distancia  
Organismo Internacional

Mérida, David; Fabregat, Ramón; Baldiris, Silvia  
SISTEMAS HETEROGÉNEOS ADAPTATIVOS BASADOS EN EL CONTEXTO  
RIED. Revista Iberoamericana de Educación a Distancia, vol. 13, núm. 2, -diciembre, 2010, pp. 73-105  
Asociación Iberoamericana de Educación Superior a Distancia  
Madrid, Organismo Internacional

Disponible en: <http://www.redalyc.org/articulo.oa?id=331427213005>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

## **SISTEMAS HETEROGÉNEOS ADAPTATIVOS BASADOS EN EL CONTEXTO**

**(DELIVERY CONTEXT-AWARE ADAPTATIVE HETEROGENEOUS SYSTEMS)**

David Mérida

Ramón Fabregat

Silvia Baldiris

Universitat de Girona (España)

### **RESUMEN**

Actualmente, la gran cantidad de tipos de dispositivos que han ganado acceso a la red es grande y diversa. Las diferentes capacidades y características de éstos, sumado a las diferentes características y preferencias del usuario, han generado un nuevo objetivo a superar: cómo adaptar los contenidos teniendo en cuenta esta heterogeneidad, conocida como el “contexto de entrega”. Los conceptos de adaptación y accesibilidad han sido ampliamente tratados y han dado como resultado gran cantidad de propuestas, técnicas y estándares tendientes a resolver el problema, haciéndose necesario acotar el análisis del objeto a considerar en el proceso de adaptación. En este trabajo, presentamos un recorrido de las diferentes propuestas y estándares que han marcado el área de trabajo de los sistemas heterogéneos y otros que han trabajado desde la interacción en tiempo real a través de plataformas basadas agentes. Todos apuntados a resolver un objetivo común: el contexto de entrega.

**Palabras clave:** sistemas hipermedia adaptativos, heterogeneidad, contexto entrega, CC/PP, independencia del dispositivo.

### **ABSTRACT**

Currently, many types of devices that have gained access to the network is large and diverse. The different capabilities and characteristics of them, in addition to the different characteristics and preferences of users, have generated a new goal to overcome: how to adapt the contents taking into account this heterogeneity, known as the “delivery context.” The concepts of adaptation and accessibility have been widely discussed and have resulted in many proposals, standards and techniques designed to solve the problem, making it necessary to refine the analysis of the issue to be considered in the process

of adaptation. We present a tour of the various proposals and standards that have marked the area of heterogeneous systems works, and others who have worked since the real-time interaction through agents based platforms. All targeted to solve a common goal: the delivery context.

**Keywords:** adaptive hypermedia systems, heterogeneity, delivery context, CC/PP, device independence.

No hace tantos años, la única forma de acceso a la web estaba relacionada a los ordenadores de escritorio o portátiles personales. Sin embargo, el explosivo incremento de dispositivos con acceso a la web planteó un nuevo objetivo: el acceso universal independiente del cómo o quién accede.

Así planteado, podría pensarse que la problemática no es compleja, pero la realidad dista mucho de este pensamiento. El concepto de “acceso universal independiente del cómo o el quién accede” ha provocado la atomización del problema y se ha reflejado en diversas propuestas para resolver la accesibilidad a la web. Propuestas que destacan un acceso universal independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios. Es decir, independiente de lo que dio por llamarse el “contexto de entrega”, o Delivery Context en su acepción en inglés.

Ahora bien, dichas consideraciones sobre la “heterogeneidad” de usuarios/dispositivos existentes han provocado el surgimiento de propuestas tales como la de la W3C en su Web Accessibility Initiative (WAI) que indica en sus premisas que “la idea principal radica en hacer una web más accesible para todos los usuarios independientemente de las circunstancias y los dispositivos involucrados a la hora de acceder a la información”.

Partiendo de esta idea, una página accesible lo será tanto para una persona con discapacidad como para cualquier otra persona que se encuentre bajo circunstancias externas que dificulten su acceso a la información (en caso de ruidos externos, en situaciones donde nuestra atención visual y auditiva no estén disponibles, pantallas con visibilidad reducida, etc.). Denotando también la necesidad de considerar los dispositivos de acceso a esta información.

Para hacer el contenido Web accesible se han desarrollado las denominadas Web Content Accessibility Guidelines 1.0 (WCAG) cuya función principal es guiar el diseño de páginas Web hacia un diseño accesible que reduzca de esta forma barreras al acceso de la información.

Por otra parte, el Device Independence Work Group2 (DIGW) de la W3C, en su trabajo sobre la “Independencia de Dispositivo” (Device Independence, 2001), plantea la premisa que independientemente del dispositivo usado para acceder a la información debe estar siempre disponible y accesible para el usuario. Es decir, tener una web universal y accesible para cualquier persona, en cualquier sitio, en cualquier momento; usando cualquier dispositivo y evitando la fragmentación de la web en espacios accesibles sólo por dispositivos concretos. Para ello, realiza un doble enfoque de la problemática, desde el punto de vista del usuario que implica un acceso universal y desde el punto de vista del desarrollador, que implica un único desarrollo con multitud de aplicaciones.

En el 2007 estos trabajos fueron potenciados y expandidos a otros niveles de abstracción a través del Ubiquitous Web Applications Working Group [UWAWG]. La nueva visión incluía la capacidad de disponer de servicios con valor añadido y modelos de negocio para dispositivos de red ubicuos. Incluyendo dentro de su actividad a los grupos Geolocation Working Group, Ubiquitous Web Applications Working Group y Device APIs and Policy Working Group.

Uno de los resultados más importantes a destacar, fue la definición de una ontología denominada Delivery Context Ontology para modelar las características del entorno en el cual los dispositivos interactúan con la web y otros servicios [DCO].

En este contexto de entrega se incluyen las características del dispositivo, el software utilizado para acceder al servicio y las características de la red que interconecta al cliente y al servidor.

Por otra parte, es un hecho que usualmente los servidores web no tienen en cuenta la diversidad y heterogeneidad de las preferencias de usuarios ni los tipos y características de sus dispositivos de acceso. Tradicionalmente se presupone que estos servidores no son responsables de proveer contenidos adaptados a dichas capacidades y/o preferencias.

Dada la gran variedad y diversidad de las capacidades de entrada/salida de estos dispositivos de acceso se hace más compleja la misión de obtener contenidos adaptados.

Resumiendo, si pensamos en la simple situación emergente en una comunicación web cliente-servidor (Figura 1) podemos identificar las primeras consideraciones y necesidades de este problema:

- herramientas que permitan disponer de las características del dispositivo de acceso,
- métodos que permitan definir las preferencias del usuario,
- protocolos de transporte de los perfiles de identificación de usuario/dispositivo,
- herramientas de edición para la creación de sitios y aplicaciones web que soporten la independencia del dispositivo de acceso,
- métodos de adaptación de contenidos en tiempo real,
- etc.

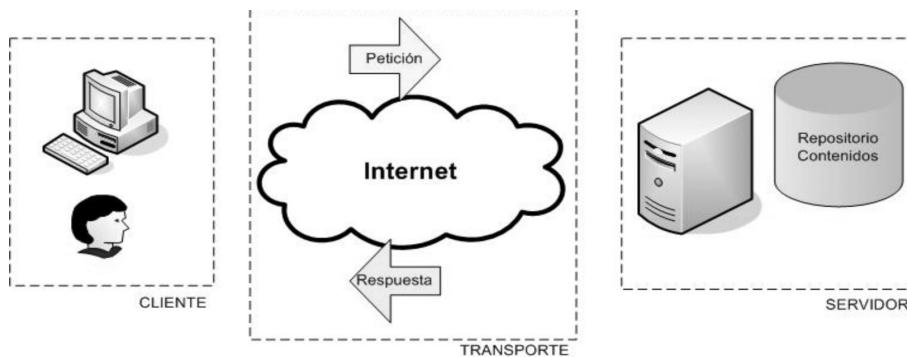


Figura 1. Comunicación web estándar cliente-servidor

Lo anteriormente expuesto pone de manifiesto cuán compleja y ambiciosa resulta ser la idea de un acceso universal independiente del cómo o quién accede.

A lo largo de este artículo, iremos explicando diversas tecnologías y propuestas para solventar esta problemática. Describiremos la necesidad de acotar el “qué y/o quién” se modela para poder adaptar los contenidos; analizaremos los distintos puntos de atención dentro del esquema de comunicación cliente-servidor; y, finalmente, circunscribiremos nuestro entorno de análisis al contexto de entrega y su afectación e implicación en las distintas técnicas de adaptación de contenidos propuestas.

Después, explicaremos qué entendemos por adaptabilidad y adaptatividad.

Para concretar qué es lo que vamos a modelar, a continuación se presentará una plataforma de usuario en la que se pueden considerar diferentes lugares para realizar la adaptación, y distintas tecnologías propuestas, surgidas a partir de estándares propuestos desde grupos de trabajo de la W3C para hacer la adaptación en el lado del cliente o en el servidor. Finalmente se presenta la arquitectura MAS-SHAAD que tiene en cuenta la heterogeneidad de usuarios y dispositivos existentes para realizar la adaptación de contenidos web.

## ADAPTABILIDAD – ADAPTATIVIDAD

En este punto vamos a rescatar los conceptos sobre los que se han basado los sistemas hipermedia adaptivos, conceptos que giran en torno a la adaptabilidad o la adaptatividad de los contenidos

El término adaptatividad ha sido muy utilizado por distintos autores en diferentes ámbitos. Dependiendo del entorno considerado, varían los objetivos perseguidos por tal adaptación.

Abdelzaher considera la adaptación de contenidos web como un mecanismo para mejorar el rendimiento de un servidor sobrecargado. En Brusilovsky (2001) se define un Sistema Hipermedia Adaptativo y se utiliza un modelo de objetivos, preferencias y conocimientos, que se construye para cada usuario, para adaptar los contenidos a sus necesidades.

De igual manera, son diferentes las clasificaciones realizadas para los sistemas hipermedia. Así, en Oppermann (1997) se plantea que el concepto de adaptación es una de las cuestiones de mayor relevancia en los sistemas de aprendizaje. Tales investigaciones demostraron que la aplicación de mecanismos de adaptación pueden proveer mejores entornos de aprendizaje en dichos sistemas hipermedia. En Opperman (1994) dos clases de sistemas fueron desarrollados para dar soporte al usuario en sus tareas:

- **Sistemas Adaptables:** permiten al usuario cambiar ciertos parámetros del sistema y adaptar el comportamiento de dicho sistema consecuentemente.
- **Sistemas Adaptativos:** Se adaptan al usuario automáticamente basándose en las suposiciones que el sistema realiza de las necesidades del usuario.

Entre esos dos extremos, se pone de manifiesto el espectro del concepto de adaptación para sistemas de computadores. Si bien es cierto que el espectro cubre un amplio rango de posibilidades existentes para el concepto de adaptación, encontramos mucho más adecuada la caracterización realizada en De Bra (1999) al referirse a los sistemas hipermedia adaptativos. En este trabajo se tienen en cuenta las preferencias del usuario como variable que decide la adaptación y se clasifican a los entornos hipermedias o a los sitios web construidos con la capacidad de realizar algún tipo de personalización en:

- Hipermedias adaptables: sistemas en los que el usuario puede proveer algún perfil (por ejemplo a través de cuestionarios) para que el sistema pueda proveer una versión de la aplicación hipermedia teniendo en cuenta ese perfil. Esta configuración, por parte del usuario, podría incluir preferencias de presentación (colores, tipo de media, estilos de aprendizaje, etc.) y el background del usuario (calificaciones, conocimiento relativo a los conceptos, etc.).
- Hipermedias adaptativos: sistemas que monitorizan el comportamiento de los usuarios y adaptan la presentación teniendo en cuenta dicho comportamiento. En estos casos la evolución, tanto en preferencias como en conocimientos, podría ser captada o deducida por el sistema a partir de los accesos realizados a las páginas. Por otra parte, muchas veces podrían necesitar cuestionarios o test a fin de obtener información más fiable respecto a las preferencias del usuario. La mayor parte de la adaptación es realizada, sin embargo, sobre la base de las acciones de navegación del usuario y también considerando el comportamiento de otros usuarios.
- Hipermedias dinámicos: el comportamiento de los usuarios es monitorizado como en los sistemas hipermedia adaptativos. En este caso, la adaptación, en lugar de ser cambiada seleccionando presentaciones predefinidas, es generada a partir de unidades atómicas de información. Es decir, es reconstruida dinámicamente a partir de los objetos individuales que componen la página tomando los que son más adecuados en base a las características del usuario.

En el presente trabajo, nos basaremos en esta clasificación por considerarla la más adecuada y representativa de nuestras caracterizaciones.

## ¿QUÉ MODELAMOS?

La necesidad de acotar el entorno de adaptación es evidente dado lo anteriormente expuesto. Por este motivo, la primera consideración queharemos es tener presente dos grandes categorías como objetos de la adaptación: el usuario y el contexto de entrega.

En este sentido en Brusilovsky y Millán (2007) se realiza un interesante planteamiento sobre “modelo del usuario” realizándose en términos generales el análisis a través de tres capas:

- Qué debe ser modelado (naturaleza).
- Cómo es representada dicha información (estructura y representación).
- Cómo son mantenidos los diferentes modelos (metodología para el modelado del usuario).

En lo concerniente a la naturaleza de la información a ser modelada, en los sistemas web adaptativos plantean la distinción entre modelos que representan:

- las características del usuario como individuo (modelado del usuario), y
- el actual contexto de trabajo del usuario (modelado del contexto).

El modelado del usuario tiene una gran importancia y ha sido considerado en todos los sistemas web adaptativos. El modelado del contexto ha sido objeto de consideración en la mayoría de sistemas adaptativos para dispositivos móviles y/o ubicuos.

El gran interés que han despertado los sistemas para dispositivos móviles y ubicuos ha generado una gran atención hacia otros campos del “contexto” tales como: ubicación, entorno físico, contexto social, estado emocional, etc. (Figura 2)

Es importante resaltar tal como se señala en Brusilovsky y Millán (2007) que el modelado del contexto es conceptualmente diferente del modelado de cualquier otra característica del usuario. No obstante, ambos modelos están estrechamente interconectados pues:

- muchos modelos de usuarios incluyen características del contexto,
- técnicas similares son usadas tanto para modelado del contexto como del usuario,
- marcos de trabajo integrados han sido desarrollados para modelar ambos, etc.

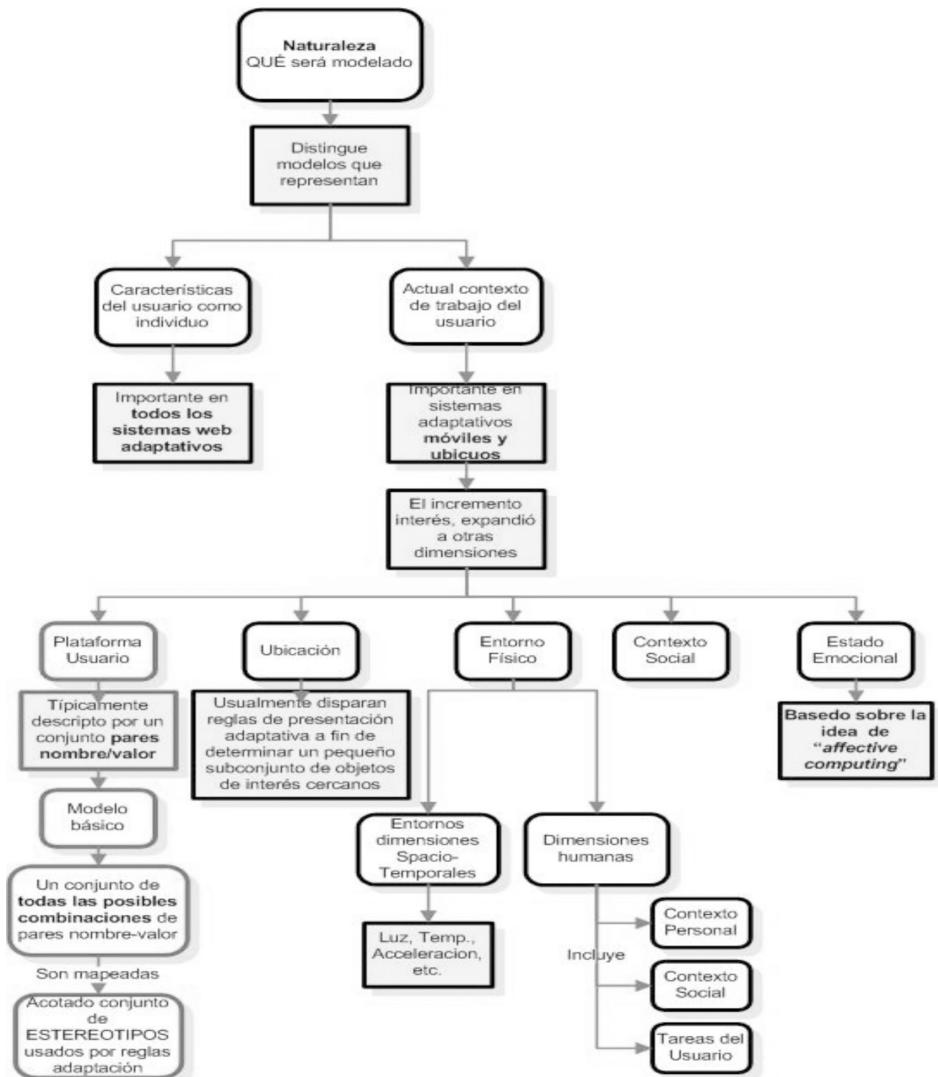


Figura 2. Capa 1: Naturaleza - ¿Qué será modelado?

## LA PLATAFORMA DEL USUARIO

Desde el punto de vista de la plataforma del usuario, el modelado del contexto involucra diversas y diferentes variables. Los sistemas hipertextuales adaptativos han explorado un amplio rango de técnicas para adaptar los contenidos a aspectos tales

como: hardware, software y el ancho de banda de conexión, etc. Como así también, técnicas tendientes a generar herramientas de edición o esquemas para la definición de los contenidos.

La Figura 3 muestra diferentes propuestas planteadas a fin de resolver el problema del modelado del contexto del usuario y de los contenidos a adaptar. Asimismo, incluye algunas propuestas utilizadas para resolver el problema de cómo realizar el transporte de los modelos de usuario.

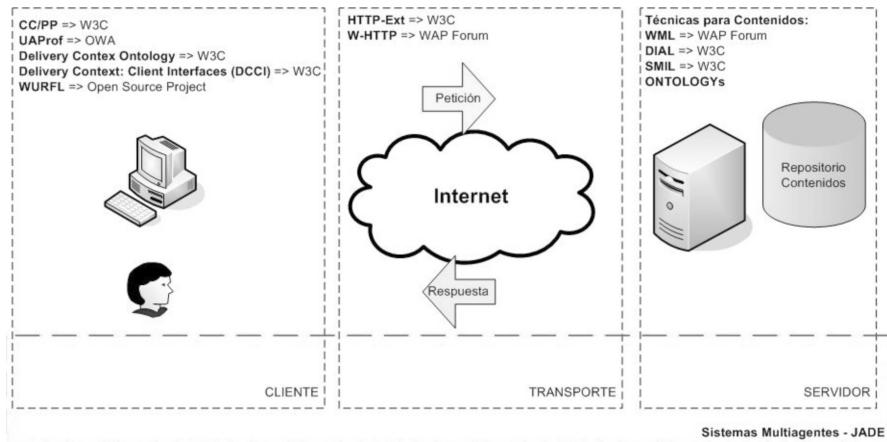


Figura 3. Propuestas para resolver el modelado del contexto de entrega

Como hemos comentado previamente, presentaremos las que se deben utilizar en el lado del cliente y las del servidor. Para el transporte de los perfiles de usuario/ dispositivo se han considerado dos protocolos: el HTTP-ex propuesto por CC/PP y el W-HTTP propuesto por UAPerf. Este último es funcionalmente idéntico a HTTP-ex pero con la característica de que es compatible con HTTP/1.1. Una explicación más detallada de estos protocolos queda fuera del alcance de este artículo.

## LADO DEL CLIENTE

En los siguientes apartados vamos a explicar cinco de las propuestas existentes que se pueden hacer en el lado del cliente: CC/PP, UAPerf, WURFL, Ontologías y DCCI.

## CC/PP – Composite Capabilities/Preferences Profile

CC/PP representa una estructura de datos que permite enviar las capacidades del dispositivo de acceso y las preferencias del usuario desde el cliente al servidor. Estos valores pueden transportarse del cliente al servidor dentro de la petición HTTP en una cabecera separada o a través de transportes definidos por otros protocolos.

El Ubiquitous Web Applications Working Group [UWAWG] de la W3C publicó en el 2007 la recomendación: Capacidad de Composición/Perfiles de Preferencia (CC/PP): estructura y vocabularios 2.0 que es una actualización de la Recomendación CC/PP 1.0 y que se ha constituido como una de las áreas de convergencia entre el W3C y la OMA3.

A través de CC/PP se define una estructura de datos y una propuesta de vocabulario para perfiles con el fin de transportar información disponible en el contexto de entrega. Basado en RDF, CC/PP representa un vocabulario neutral que permite el desarrollo e implementación de otros vocabularios por parte de la comunidad involucrada en el desarrollo de aplicaciones, dispositivos y navegadores. Dado el contenido dinámico existente en estos perfiles, estos pueden estar distribuidos en múltiples repositorios a través de la web.

CC/PP ha sido, durante un tiempo, la propuesta preferida para el intercambio de la información del contexto de entrega entre clientes, intermediarios y servidores.

### ¿Cómo describir la capacidad de dispositivos y las preferencias del usuario?

Es grande y diversa la cantidad de nuevos dispositivos con acceso a la web. Dispositivos con marcadas diferencias que reflejan: capacidades tanto de entrada como de salida, lenguajes soportados, conectividad a la red, soporte multimedia, etc. que imposibilitan muchas veces la correcta entrega de los contenidos disponibles. Así mismo, los usuarios al acceder a estos contenidos desearían tenerlos adaptados dinámicamente reflejando en tiempo real sus cambios de preferencias (ej. audio sí/no, vídeo sí/no, etc).

Para resolver el problema se hace necesario codificar en el lado del cliente este contexto de entrega, de tal manera que el servidor pueda utilizarlo para personalizar los contenidos teniendo en cuenta las características del dispositivo y del usuario.

CC/PP utiliza, para definir el contexto, una estructura simple y jerarquizada en dos niveles: componentes y pares de atributo/valor. Un componente representa una parte del contexto de entrega (ej. características de la red, el software soportado o las características del hardware del dispositivo) y puede contener uno o más atributos. En el siguiente ejemplo se muestra un componente que codifica las preferencias del usuario (user preferences) y que contiene, entre otros, el atributo que especifique si desea o no las características de audio (AudioOutput).

```
<ccpp:component>
  <rdf:Description rdf:ID="UserPreferences">
    <rdf:type rdf:resource="http://www.example.org/profiles/prefs/v1_o#User
    Preferences"/>
    <ex:AudioOutput>Yes</ex:AudioOutput>
    <ex:Graphics>No</ex:Graphics>
    <ex:Languages>
      <rdf:Seq>
        <rdf:li>en-cockney</rdf:li>
        <rdf:li>en</rdf:li>
      </rdf:Seq>
      </ex:Languages>
    </rdf:Description>
  </ccpp:component>
```

Debe remarcarse que CC/PP no define un vocabulario específico ni un protocolo de transporte para estas instancias de vocabulario CC/PP que también son llamados perfiles.

### **UAPerf – User Agent Profiles**

El Open Mobile Alliance (OMA) definió UAPerf (2001) como una implementación de CC/PP para dispositivos móviles. Al igual que CC/PP, un perfil UAPerf define una jerarquía de dos niveles integrada por: componentes y atributos.

A diferencia de CC/PP, la especificación UAPerf propone un vocabulario (un conjunto específico de componentes y atributos) para describir la generación de teléfonos WAP4 y describe dos protocolos para la transmisión del perfil desde el cliente al servidor. Los perfiles usados por el vocabulario UAPerf están integrados de

seis componentes: HardwarePlatform, SoftwarePlatform, NetworkCharacteristics, BrowserUA, WapCharacteristics y PushCharacteristics, cada uno de los cuales contiene atributos. Cada atributo posee un nombre distintivo, un tipo de colección asociada, un tipo de atributo y una regla de resolución.

En UAProf existen tres tipos de colección:

- Simple: contiene un valor simple (Ej. ColorCapable en el componente HardwarePlatform).
- Bag: contiene valores múltiples desordenados (Ej. BluetoohProfile en el componente HardwarePlatform).
- Seq: contiene valores múltiples ordenados (Ej. Ccpp-AcceptLanguage en el componente SoftwarePlatform).

A su vez, cada atributo puede tener uno de los cuatro tipos de atributos:

- String (Ej. BrowserName en BrowserUA).
- Boolean (Ej. ColorCapable en HardwarePlatform).
- Number: es un entero positivo (Ej. BitsPerPixel en HardwarePlatform).
- Dimension: es un par de enteros positivos (Ej. ScreenSize en HardwarePlatform).

Finalmente, los atributos están asociados con una regla de resolución:

- Locked indica que el valor final de un atributo es la primera ocurrencia del atributo fuera del bloque de descripción por defecto.
- Override indica que el valor final de un atributo es la última ocurrencia del atributo fuera del bloque de descripción por defecto.
- Append indica que el valor final del atributo es la lista de todas las ocurrencias del atributo fuera del bloque de descripción por defecto.

El siguiente ejemplo muestra un fragmento del componente Hardware Platform de UAPerf que define las características de la pantalla y bluetooth del dispositivo:

```
<prf:component>
<rdf:Description rdf:ID="HardwarePlatform">
<rdf:type rdf:resource="http://www.openmobilealliance.org/profiles/
UAPROF/ccpps schema-20021113#HardwarePlatform"/>
<prf:ScreenSizeChar>15x6</prf:ScreenSizeChar>
<prf:BitsPerPixel>2</prf:BitsPerPixel>
<prf:ColorCapable>No</prf:ColorCapable>
<prf:BluetoothProfile>
<rdf:Bag>
<rdf:li>headset</rdf:li>
<rdf:li>dialup</rdf:li>
<rdf:li>lanaccess</rdf:li>
</rdf:Bag>
</prf:BluetoothProfile>
</rdf:Description>
</prf:component>
```

El estándar WAP 1.2.1 recomienda el transporte de la información UAPerf sobre Internet utilizando HTTP Extension Framework (HTTPEx, 2000) que originalmente fue sugerido por CC/PP. Debido a la falta de implementación de HTTPEx, WAP 2.0 definió en su lugar la extensión de HTTP 1.1 como un protocolo de Internet en el que utilizó cabeceras personalizadas. Este protocolo puede soportar tanto perfiles estáticos como dinámicos.

Un perfil estático es accedido a través de una dirección web o URI. Esto conlleva algunas ventajas, ya que una petición desde un cliente a un servidor contendrá únicamente dicha URI y no un extenso documento XML. El cliente no tendrá que cargar ni crear el perfil y por lo tanto no se carga la comunicación con el servidor.

En cambio un perfil dinámico es generado en tiempo real (on-the-fly) y consecuentemente no tiene asociado una URI. El perfil consiste en un fragmento de perfil contenido la diferencia respecto del perfil estático, e incluso podría contener datos únicos no incluidos en el perfil estático.

UAProf ha tenido una amplia repercusión, incorporándose entre otros proyectos a Deli (DELI, 2006). Finalmente, cabe destacar la existencia de numerosos repositorios UAProf disponibles aún para su consulta, siendo uno de los más importantes el mantenido por la W3C5.

### **WURFL - Wireless Universal Resource File**

WURFL (Passani, 2010) es un proyecto de código abierto dentro del marco de SourceForge.net6 que provee una fuente alternativa de información a UAProf y está dirigido al amplio espectro de desarrolladores de aplicaciones y servicios de entornos WAP e inalámbricos.

La base de datos de WURFL es esencialmente un gran fichero XML que contiene información sobre las capacidades y características de miles de dispositivos móviles y sus variantes.

Estos datos están organizados jerárquicamente y son específicos de un dispositivo y de una o más familias del dispositivo. Todos estos datos son mapeados a través de un atributo/mecanismo recursivo denominado “fall-back” que nos permite con poco esfuerzo construir una amplia lista de capacidades de los dispositivos.

El principal alcance de este fichero XML es recolectar toda la información posible sobre la totalidad de dispositivos móviles existentes con acceso a páginas WAP para facilitar a los desarrolladores la construcción de mejores aplicaciones y servicios para los usuarios.

Toda esta información ha sido recolectada por gran cantidad de personas de diferentes países y son de libre utilización en cualquier tipo de aplicación, comercial o no. La única condición requerida es dejar pública cualquier modificación realizada sobre dicho fichero, continuando de esta manera con el espíritu original e idea de los creadores del proyecto.

### **¿Cómo funciona?**

En el ejemplo se muestra un dispositivo cuyo agente de usuario/navegador (user\_agent) se identifica asimismo como “SEC-SGHE950/1.0 NetFront/3.4 Profile/MIDP-2.0 Configuration/CLDC-1.1”. Consultando WURFL encontramos lo siguiente:

```
<device user_agent="SEC-SGHE950/1.0 NetFront/3.4 Profile/MIDP-2.0
    Configuration/CLDC-1.1"
    fall_back="samsung_sgh_e950_ver1"
    id="samsung_sgh_e950_ver1_sub1"/>;
```

La respuesta WURFL no dice mucho a primera vista y tan solo podríamos asumir que WURFL tiene identificado al dispositivo o familia de dispositivo a través de su agente de usuario. Sin embargo, el atributo de “fall\_back” cobra gran importancia en este momento ya que indica que el dispositivo pertenece también a la familia de dispositivos identificados como “samsung\_sgh\_e950\_ver1”. Al preguntar a WURFL por los detalles de este ítem se obtiene:

```
<device user_agent="SEC-SGHE950"
    actual_device_root="true"
    fall_back="sec_e900_ver1"
    id="samsung_sgh_e950_ver1">
<group id="product_info">
    <capability name="model_name" value="E950"/>
</group>
<group id="display">
    <capability name="resolution_width" value="240"/>
    <capability name="resolution_height" value="320"/>
    <capability name="max_image_width" value="233"/>
    <capability name="max_image_height" value="280"/>
</group>
<group id="markup">
    <capability name="preferred_markup" value="html_wi_oma_xhtmlmp_1_0"/>
</group>
</device>
```

Con esta respuesta hemos identificado el dispositivo, su modelo (E950), su tamaño de pantalla (240x320) y que aceptaría preferentemente páginas web en formato XHTML-MP. Sin embargo, podemos ir más lejos y nuevamente con el atributo “fall\_back” registrado y apuntando a “sec\_e900\_ver1” obtendríamos:

```
device user_agent="SEC-SGHE900"
    actual_device_root="true"
    id="sec_e900_ver1"
    fall_back="netfront_ver3">
<group id="product_info">
    <capability name="brand_name" value="Samsung"/>
    <capability name="model_name" value="E900"/>
</group>
<group id="markup">
<capability name="preferred_markup" value="html_wi_oma_xhtmlmp_1_0"/>
    <capability name="html_wi_oma_xhtmlmp_1_0" value="true"/>
    <capability name="html_wi_w3_xhtmlbasic" value="true"/>
    <capability name="wml_1_3" value="true"/>
</group>
<group id="display">
    <capability name="resolution_width" value="240"/>
    <capability name="resolution_height" value="320"/>
    <capability name="max_image_height" value="300"/>
    <capability name="max_image_width" value="232"/>
</group>
<group id="image_format">
<capability name="gif" value="true"/>
    <capability name="jpg" value="true"/>
    <capability name="png" value="true"/>
    <capability name="colors" value="262144"/>
</group>
<group id="storage">
    <capability name="max_deck_size" value="8000"/>
</group>
<group id="object_download">
    <capability name="ringtone" value="true"/>
    <capability name="ringtone_midi_monophonic" value="true"/>
    <capability name="ringtone_midi_polyphonic" value="true"/>
<capability name="ringtone_aac" value="true"/>
<capability name="ringtone_mp3" value="true"/>
<capability name="wallpaper" value="true"/>
<capability name="wallpaper_gif" value="true"/>
<capability name="wallpaper_jpg" value="true"/>
<capability name="screensaver" value="true"/>
```

```
<capability name="screensaver_gif" value="true"/>
<capability name="video" value="true"/>
<capability name="video_qcif" value="true"/>
<capability name="video_sqcif" value="true"/>
<capability name="video_wmv" value="true"/>
<capability name="video_3gpp" value="true"/>
<capability name="video_mp4" value="true"/>
<capability name="video_acodec_aac" value="true"/>
<capability name="wallpaper_colors" value="18"/>
<capability name="wallpaper_png" value="true"/>
<capability name="wallpaper_preferred_height" value="320"/>
<capability name="wallpaper_preferred_width" value="240"/>
<capability name="ringtone_VOICES" value="64"/>
</group>
<group id="sound_format">
    <capability name="aac" value="true"/>
    <capability name="midi_monophonic" value="true"/>
    <capability name="midi_polyphonic" value="true"/>
    <capability name="mp3" value="true"/>
    <capability name="voices" value="64"/>
</group>
<group id="j2me">
    <capability name="j2me_midp_2_0" value="true"/>
    <capability name="j2me_midp_1_0" value="true"/>
    <capability name="j2me_cldc_1_1" value="true"/>
</group>
</device>
```

A través de la técnica de “fall\_back” hemos conseguido mucha información del dispositivo y podríamos continuar haciéndolo preguntando hacia atrás considerando sucesivamente los valores “netfront\_ver3”, “generic\_xhtml”, “generic” y finalmente “root” y obteniendo finalmente toda la información disponible para este dispositivo.

El mayor inconveniente de la propuesta de WURFL es la magnitud del fichero XML de consulta resultante. Miles y miles de líneas, en un fichero único, que a la hora de lanzar consultas recursivas en tiempo real, pueden llegar a representar una opción poco práctica. Para superar este problema WURFL posibilita la utilización

de esta gran base de datos en local y no solamente a través de un único fichero disponible en la red.

## Ontologías

A partir del 2007, el grupo Ubiquitous Web Application (UWA) de la W3C se hizo cargo de los trabajos destinados a la definición de servicios con valor añadido y modelos de negocios para dispositivos ubicuos, tarea hasta entonces portada por el Device-Independence Work Group<sup>2</sup>. De sus actividades surge la ontología del contexto de entrega, a través de la cual se provee un modelo formal de las características del entorno en el que interactúan los dispositivos con la web u otros servicios.

Así, mientras la ontología captura el contexto de uso, en el cual un usuario está interactuando, el contexto de entrega incluye las características del dispositivo, del software utilizado para acceder a los servicios y de la red que provee la conexión.

La ontología está formalmente especificada en y (OWL, 2009) y define un vocabulario normativo de términos (clases, propiedades e instancias) que modelan las diferentes propiedades, aspectos y componentes del contexto de entrega.

El modelo representado por la ontología es esencialmente jerárquico. En la raíz de dicha jerarquía se posiciona la clase *Delivery Context* que da acceso a los elementos *UserAgent*, *NetworkBearer*, *Device*, *Runtime Environment* y *Environment* (entorno físico) que son elementos esenciales de cualquier contexto de entrega. Cada uno de estos elementos está representado por clases que, a su vez, tienen propiedades que modelan sus características específicas y sus componentes.

Modularmente, la ontología ha sido dividida en nueve grupos de términos que representan el contexto de entrega:

- Common: para propósitos generales, clases genéricas, propiedades e instancias que soportan el modelado de los diferentes aspectos del contexto de entrega (Ej. protocolo nivel aplicación, idioma, ancho/alto de resolución, código del país, etc.)
- Hardware: propiedades, clases e instancias que modelan los elementos de hardware de un contexto de entrega (Ej. soporte de bluetooth, cámara, memoria, CPU, etc.)
- Java: estos términos modelan un mínimo conjunto de propiedades y clases que permiten representar las características relacionadas con el soporte de las tecnologías Java (Ej. entorno de ejecución, plataforma, configuración, etc.)

- Location: este grupo está dirigido a definir el conjunto de clases, propiedades e instancias que tienen relación con la ubicación geográfica (Ej. coordenadas UTM, altitud, latitud, longitud, etc.).
- Main: representan los Aspectos y Propiedades fundamentales del dispositivo o el contexto de entrega en sí mismo (Ej. actuales dispositivos en uso, red disponible, navegador, etc.).
- Network: este grupo modela clases, propiedades e instancias relacionadas con las características de la red (Ej. soporte red cableada, wireless, proxy, ancho de banda, etc.).
- Push: representa el soporte de mensajería MMS, WAP y tecnología de push-request.
- Software: modelan los elementos de software (Ej. formato de audio y video soportados, tipo de reproductor de audio, reproductor de video, sistema operativo, etc.).
- Web Browsing: modela un conjunto mínimo de propiedades y clases que permiten representar el soporte de navegadores web.

Como hemos dicho, esta ontología tiene como objetivo proveer un modelo formal y universalmente aceptado para el contexto de entrega. No pretende modelar propiedades que puedan ser dependientes de una aplicación en particular.

### **DCCI – Delibery Context Client Interfaces**

En (DCI, 2006) y (DCI, 2007) se define una plataforma y una interfaz de programación de lenguaje natural para aplicaciones web, para proveerles acceso a la estructura jerárquica de propiedades dinámicas que representan las capacidades del dispositivo, configuraciones, preferencias de usuario y condiciones de entorno de trabajo del usuario.

Muchas aplicaciones, particularmente aquellas independientes del dispositivo, esperan funcionar en entornos heterogéneos con una amplia variedad de capacidades. La configuración del dispositivo, las preferencias del usuario y condiciones de entorno pueden variar en tiempo real y estas aplicaciones necesitan ser capaces de adaptar consecuentemente los contenidos.

Así por ejemplo, ¿cómo podría una aplicación negociar con situaciones tales como: alertas de batería baja, o variaciones de las capacidades de la red, o (peor aún) la pérdida de conectividad? Tal vez el usuario podría silenciar su micrófono o deshabilitar la salida de audio durante una sesión. Estas configuraciones/

propiedades dinámicas, que por lo general son temporales, reflejan situaciones que alteran el contexto de entrega de los contenidos y por lo tanto deberían ser consideradas.

Ahora bien, es importante distinguir entre propiedades persistentes (estáticas) y transitorias (dinámicas). Las propiedades estáticas se refieren a datos que permanecen constantes durante la sesión (Ej. el idioma seleccionado, tipo y tamaño de fuentes, etc.); mientras que las propiedades dinámicas se refieren a notificaciones y eventos durante la sesión (Ej. notificaciones de cambio de posición generados por un GPS, variaciones de la capacidad de la red, etc.).

DCCI provee métodos de acceso para gestionar estas propiedades estáticas y dinámicas de manera que el contenido pueda ser adaptado al contexto particular de entrega. Existen dos mecanismos generales para acceder a dicho contexto: uno basado en la especificación de interfaces de programación; y, el otro basado en representaciones específicas del contexto de entrega.

## **Interfaces de programación**

Las interfaces proveen una forma de acceder al contexto de entrega independientemente de la forma en que la información esté almacenada y representada.

Existen hoy en día, por parte de los fabricantes de dispositivos, un gran número de representaciones propietarias del contexto de entrega, que dificultan el trabajo de los desarrolladores al tener que considerarlos. La utilización de interfaces estandarizadas podrían proveer mecanismos comunes que les posibilitaría obtener independencia de la representación particular usada.

La disponibilidad de las propiedades del contexto de entrega ha sido focalizada por las interfaces DCCI de diferentes maneras. Algunas posibilitando su disponibilidad al código que se ejecuta en el cliente; poniendo dicha información disponible en el lado del servidor; haciendo énfasis más en la información estática y otras por sus propiedades dinámicas. Existe también un conjunto de interfaces específicamente orientadas a proveer acceso al contexto de entrega directamente desde los lenguajes etiquetados.

## **Representaciones**

Previo a la definición de interfaces, los trabajos se avocaron a la estandarización de la definición de representaciones de las propiedades del contexto de entrega. Uno

de estos estándares es CC/PP. Como se expuso anteriormente, CC/PP provee un marco de trabajo para la representación del contexto de entrega. Basado en RDF, no especifica en sí mismo un vocabulario particular y por lo tanto, no provee una representación del contexto de entrega. Sin embargo, provee de un marco de trabajo en el cual las representaciones pueden ser construidas.

Por otro lado, UAProf aporta un vocabulario específico. En general, es utilizado para la representación de las propiedades estáticas y está asociado normalmente con direcciones URI que permiten acceder a los perfiles y analizarlos a fin de recuperar la información de las propiedades en ellos contenidas.

## **LADO DEL SERVIDOR /REPOSITORIO DE CONTENIDOS**

En el lado del servidor, y más específicamente en el lado del repositorio de contenidos, los esfuerzos para resolver la problemática del contexto de entrega se han reflejado en diferentes propuestas de lenguajes de etiquetado, tales como WML, DIAL y SMIL.

### **WML - Wireless Markup Language**

En WML (2000) y WML (2001) se define un lenguaje de marcas comprendido dentro del estándar XML 1.0 (eXtensible Markup Language) que es utilizado para la construcción de las páginas de dispositivos móviles. Propuesto por la WAP4 Forum (actualmente OMA3), WML es un metalenguaje que además de usar etiquetas predefinidas nos permite crear componentes propios.

Las páginas WML a menudo se denominan “decks” (barajas). Cada deck contiene un conjunto de “cards” (cartas). Un elemento carta puede contener texto, marcas, enlaces, campos de entrada, imágenes, etc. Y estas cartas pueden estar relacionadas unas con otras a través de enlaces.

Cuando una página WML es accedida desde un dispositivo móvil, todas las cartas en la página son descargadas por el dispositivo desde el servidor WAP. En ese momento, la navegación entre cartas se realiza directamente sobre el dispositivo, internamente, sin ninguna intervención extra por parte del servidor.

En el siguiente ejemplo, se muestra un ejemplo del etiquetado básico de una página WML.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="no1" title="Card 1">
    <p>Hello World!</p>
  </card>

  <card id="no2" title="Card 2">
    <p>Welcome to our WAP Tutorial!</p>
  </card>
</wml>
```

### **DIAL - Device Independent Authoring Language**

Desde hace tiempo, el DIWG2 trabaja en diferentes técnicas destinadas a solucionar el problema de la edición/creación (authoring) de contenidos y aplicaciones multidispositivo. En particular, se resaltan dos aspectos importantes dentro de esta área.

El primero es un estándar desarrollado para la autoría de contenidos, independiente del dispositivo de acceso (DIAL, 2010). El segundo está enfocado en los trabajos relacionados con las representaciones explícitas del diseño. Éstas servirían para ser utilizadas en lugar de las referencias implícitas en el lenguaje de marcado. La idea es proporcionar un mecanismo efectivo que permita la completa separación entre contenido y presentación. Entre otras opciones, se ha considerado extender las especificaciones de Cascading Style Sheets (CSS2) y (CSS3) para recoger este tipo de representación del estilo independiente del dispositivo, pero aún se encuentra en estado de desarrollo.

Por su parte, DIAL no es un lenguaje nuevo como tal sino que se fundamenta en especificaciones existentes y emergentes de la W3C: (XHTML), (Xforms) y (DISelect).

DIAL tiene como objetivo asistir a los autores/desarrolladores en los desafíos que surgen durante la creación de sitios y aplicaciones con soporte a entornos independientes del dispositivo. Desafíos que los autores comúnmente hacen frente

durante la construcción de sitios/aplicaciones que pueden ser accedidos por usuarios a través de una gran variedad de dispositivos con diferentes capacidades.

Esto se consigue permitiendo a los autores declarar “propósitos de creación” (authorial intent) que hacen referencia a las condiciones bajo la cuales el contenido debería ser elegido o filtrado. En el siguiente ejemplo, el autor desea que los usuarios que se subscriben al servicio reciban una presentación “especial” del contenido (el video HW\_movie\_portrait.mov) mientras que el resto recibirá una presentación normal (la imagen HW\_image.jpg).

```
...
<sel:select>
  <!-- check if the requesting user is a subscriber-->
  <sel:when expr="den:search('userSubscriptionStatus') = premium">
    <!-- if so select this movie -->
    <object sel:selid="HelloWorldMovie" src="HW_movie_portrait.mov"/>
  </sel:when>
  <sel:otherwise>
    <!-- if not send an image instead -->
    <object sel:selid="HelloWorldMovie" src="HW_image.jpg"/>
  </sel:otherwise>
</sel:select>
...
...
```

## **SMIL - Synchronized Multimedia Integration Language**

SMIL se constituye como el lenguaje de integración de multimedia sincronizado. Desarrollado por la W3C desde 1998, este lenguaje permite integrar un amplio conjunto de objetos multimedia independientes, dentro de una presentación multimedia sincronizada. A través de SMIL un autor puede:

- describir el comportamiento temporal de la presentación.
- describir la distribución de los elementos en la pantalla.
- crear hiperenlaces con objetos multimedia.

Basado en XML, permite a los desarrolladores mezclar distintos medios para ser presentados y sincronizados unos con otros. Fácil de utilizar, posee una integración completa con HTML y JavaScript que lo convierte en una potente herramienta para desarrolladores de entornos multimedia.

## Ontología para el repositorio de contenidos (MAS-SHAAD)

En Mérida (2007) se presenta un generador dinámico de contenidos para la adaptación en sistemas hipermedia. El mismo fue diseñado a través de una estructura modular que puede escalarse fácilmente a fin de considerar otras variables de adaptación.

El generador consta esencialmente de dos partes (Figura 4):

- un repositorio de contenidos que almacena los elementos atomizados que conforman la página web.
- un montador de contenidos que, con la ayuda de hojas de estilo CSS, permite obtener los contenidos del repositorio adaptados.

El repositorio está formado por dos elementos de software: un repositorio de instancias RDF y una base de datos relacional para indexar y almacenar los contenidos binarios.

Para los objetos que conforman las páginas web se define una ontología que describe los metadatos y la estructura que tendrán los datos a gestionar (textos de diferentes idiomas y diferentes extensiones, imágenes de diferentes resoluciones, sonidos en diferentes calidades, etc.).

Una vez obtenidos y seleccionados los contenidos dinámicamente, el montador de contenidos, utilizando hojas de estilo CSS (para darles un estilo de apariencia) transforma los objetos atomizados en contenidos XHTML y obtiene de esta forma la página web adaptada a las características del usuario final.

En este trabajo se consideraron solamente las preferencias del usuario y las características del dispositivo de acceso, pero representa una buena alternativa para el tratamiento en tiempo real de los objetos considerados durante la adaptación de contenidos.

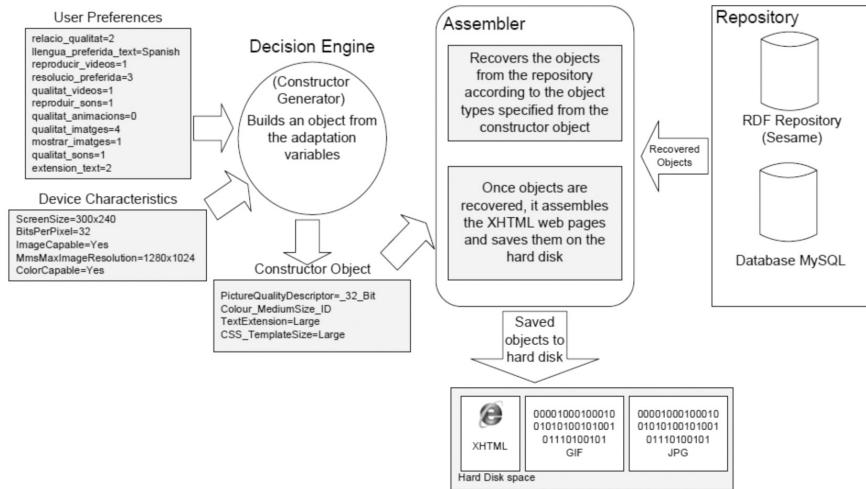


Figura 4. Generador Dinámico de Contenidos

Esta implementación fue integrada posteriormente en su totalidad dentro del sistema multiagente implementado en MAS-SHAAD (Mérida, 2007).

## MAS-SHAAD

En los capítulos anteriores hemos visto diferentes técnicas surgidas a partir de estándares propuestos a través de grupos de trabajo de la W3C (tales como DIWG, UWAWG, CC/PP WG), OMA, WAP Forum, etc. todas ellas tendientes a definir modos de representación del heterogéneo contexto de entrega existente. Sin embargo, otras propuestas han utilizado agentes inteligentes o sistemas multiagentes para solucionar este problema. Estas alternativas, solas o en combinación con otras técnicas, se han planteado como otro recurso para en los entornos heterogéneos entregar al usuario páginas web adaptadas a sus necesidades, preferencias u otras consideraciones de su contexto de entrega.

MAS-SHAAD (Merida, 2007) es una implementación multiagente del modelo SHAAD (Merida, 2002). En la figura 5 se presenta un esquema general de cómo se plantea el proceso de adaptación en el modelo SHAAD.

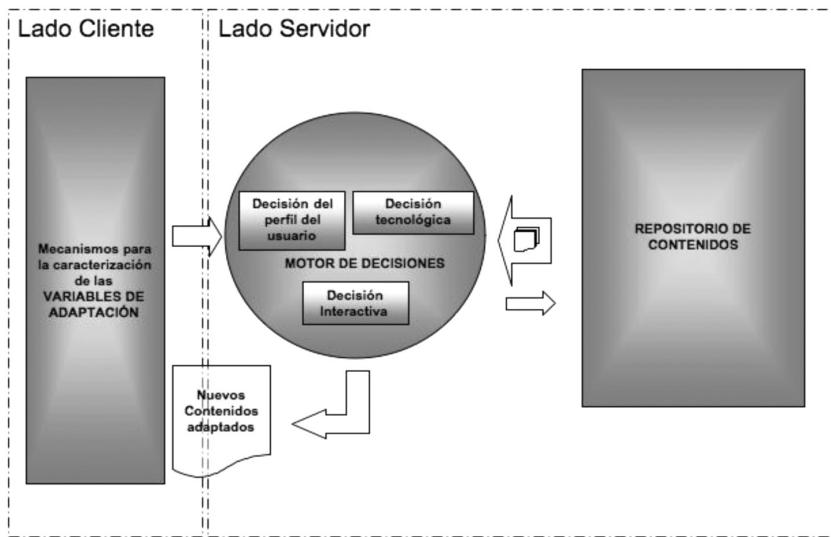


Figura 5. *Modelo General de Adaptación de SHAAD*

En el lado del cliente se realiza una caracterización de las variables de adaptación a ser consideradas; y del lado del servidor se encuentran localizados dos elementos: por una parte, el motor de decisiones que soporta la toma de la decisión de adaptación; y, por otra parte, el repositorio de contenidos, desde el cual se escogerá un subconjunto de esos contenidos que serán mostrados al usuario de acuerdo a su perfil.

MAS-SHAAD en particular está construido sobre la plataforma de JADE y utiliza CC/PP para almacenar en el Modelo de Usuario las características del dispositivo de acceso. Los clientes pueden especificar sus características haciendo referencia al perfil estándar accesible al servidor, o pueden expandir dicho perfil agregando nuevas características.

Entre el cliente y el servidor HTTP se establece una capa de comunicación suplementaria formada por el sistema de agentes que nos proporciona JADE. Estos agentes se distribuyen entre los diferentes tipos de nodos existentes de la siguiente manera (Figura 6):

- **Nodo cliente** (o usuario del sistema): solicita los contenidos y los recibe del servidor web a través del protocolo HTTP. Contiene un agente denominado “agente gestor cliente” que permite al cliente establecer los valores de las

variables necesarias para realizar la adaptación, obtener las características tecnológicas del dispositivo de acceso, obtener un identificador para el cliente y registrarla antes de que utilice el sistema.

- **Nodo servidor web:** contiene un conjunto de agentes (“agente servlet”, “agente decisor” y “agente constructor”) que recogen los valores de las variables de adaptación recopiladas por el “agente gestor cliente” y las utilizan para construir, a partir de los contenidos almacenados en el mismo servidor, las respuestas adaptadas a las necesidades del cliente. El “agente servlet” realiza el enlace entre el servlet y la plataforma de agentes del servidor web. El “agente decisor” evalúa los datos sobre el cliente que le son proporcionados y determina como debe ser la respuesta que se le debe entregar. Finalmente, el “agente constructor” recibe las indicaciones del “agente decisor” y construye la página web que le será entregada al cliente.
- **Nodo servidor de identificación y registro:** contiene un agente denominado “agente generador de identificadores” que proporciona un identificador único a cada “agente gestor cliente”.

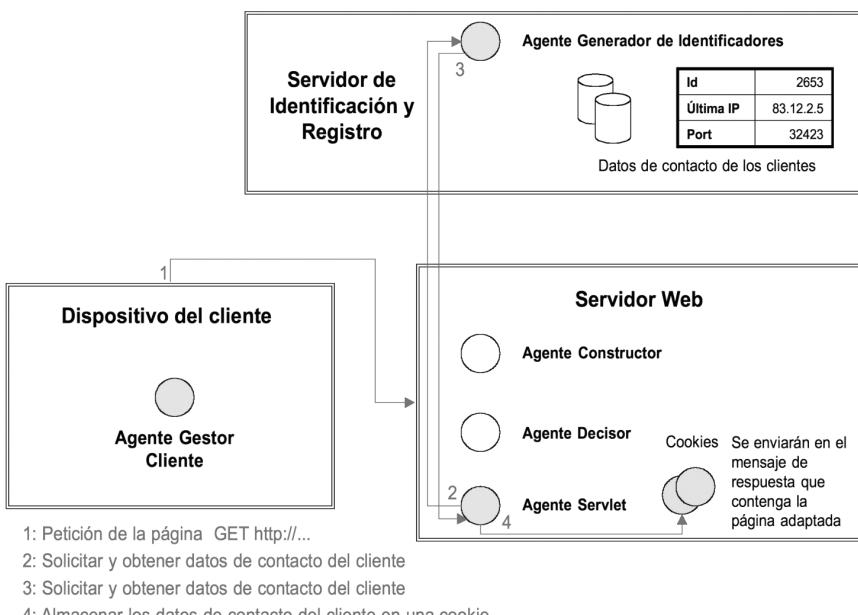


Figura 6. Arquitectura modular MAS-SHAAD

De manera resumida, el comportamiento general del sistema es el siguiente. El usuario debe registrarse antes de realizar la petición HTTP al servidor web. Una vez registrado, el sistema verifica las preferencias del usuario, obtiene sus datos y establece la comunicación con la plataforma del agente cliente. Establecida la comunicación, el agente servlet obtiene las características tecnológicas del dispositivo de acceso, ya sea consultando el repositorio de dispositivos administrado por el mismo servidor, o a través de la URL especificada por el fabricante para recuperar los datos CC/PP. Posteriormente recupera los datos y los carga en el repositorio.

Con todos estos datos, prepara los parámetros que serán pasados al agente constructor, que es el responsable de la generación de la página web para el usuario. Si por alguna razón estos datos no estuviesen disponibles, no se considerará ningún dato para ejecutar la adaptación.

## CONCLUSIONES

Hemos recorrido a lo largo de este trabajo un abanico sesgado de propuestas existentes en el campo de los sistemas heterogéneos, enfocados a solventar el problema del contexto de entrega.

Propuestas dirigidas a solventar el problema de la caracterización del cliente en términos genéricos, con el fin de poder utilizar tal información para generar una entrega adecuada de contenidos que tenga en cuenta tal caracterización.

Así, hemos visto cómo la propuesta basada en CC/PP se ofrece como una solución viable para caracterizar el dispositivo de acceso y las preferencias del usuario. Modelo sobre el que se sustentan UAProf y posteriormente WURFL.

Así también, se ha resaltado la importancia de los lenguajes de etiquetado (tales como WML, DIAL y SMIL), planteados como herramientas útiles para los desarrolladores durante la generación de contenidos.

Finalmente, hemos presentado la propuesta de MAS-SHAAD basada en una plataforma multi-agente que en combinación con CC/PP pone de manifiesto la importante sinergia obtenida en una propuesta de modelado en tiempo real.

Es de nuestro interés actual avanzar en el desarrollo de aproximaciones que permitan realizar procesos de adaptación considerando no solo el dispositivo de acceso de los usuarios, sino ampliar la visión del contexto considerando variables

como su localización espacial, el ambiente físico en el que se encuentra, las condiciones especiales de acceso identificadas, entre ellas la discapacidades físicas, las dificultades de aprendizaje, deficiencias en las funciones ejecutivas, así como la caracterización de las tecnologías asistivas a las que haya lugar.

## NOTAS

1. La “World Wide Web Consortium (W3C)” es un comunidad internacional cuyo trabajo está enfocado en el desarrollo de estándares para la Web. Liderado por su creador Tim Berners-Lee y su CEO Jeffrey Jaffe, la misión de la W3C está enfocado en la obtención del pleno potencial existente en la Web. Más información en <http://www.w3.org/>
2. El grupo de trabajo de la DIGW (Device Independence Working Group) se forma para continuar con los trabajos de la CC/PP WG (Composite Capabilities/Preference Profile Working Group - <http://www.w3.org/Mobile/CCPP/Historic-CCPP1>). Actualmente estos trabajos han sido delegados en el grupo de trabajo UWAWG (Ubiquitous Web Applications Working Group) de la W3C.
3. OMA (Open Mobil Aliance) es la organización líder en el desarrollo de estándares abiertos para la industria de telefonía móvil. Más información en <http://www.openmobilealliance.org/>
4. WAP (Wireless Application Protocol) es un estándar abierto y global con el fin de potenciar a los usuarios de dispositivos inhalámbricos, acceder e interactuar fácil e instantáneamente con la información y los servicios web. El organismo que se encarga de desarrollar el estándar WAP fue originalmente el WAP Forum, fundado por cuatro empresas del sector de las comunicaciones móviles, Sony-Ericsson, Nokia, Motorola y Openwave (originalmente Unwired Planet). Desde 2002 el WAP Forum es parte de la Open Mobile Alliance (OMA).
5. [http://w3development.de/rdf/uaprof\\_repository/](http://w3development.de/rdf/uaprof_repository/)
6. SourceForge.net es el sitio web de desarrollo de software de código libre y abierto más grande del mundo. Más información: <http://sourceforge.net/>
7. Este trabajo ha sido parcialmente financiado por proyecto A2UNA@ [TIN2008-06862-Co4-02/TSI]. Con el apoyo del programa Alban, Programa de Becas de Alto Nivel de la Unión Europea para América Latina, beca No. Eo6D103680CO.

## REFERENCIAS BIBLIOGRÁFICAS

- Allan, J.; Ford, K.; Richards, J.; Spellman, J. (2010). W3C. User Agent Accessibility Guidelines (UAAG) 2.0. W3C Working Draft. [en línea] Disponible en: <http://www.w3.org/TR/UAAG20/> (consulta 2010, 24 de abril).
- Axelsson, J.; Birbeck, M.; Dubinko, M.; Epperson, B.; Ishikawa, M.; McCarron, S.; Navarro, A.; Pemberton, S. (2006).
- W3C. XHTML™ 2.0. W3C Working Draft. [en línea] Disponible en: <http://www.w3.org/TR/2006/WD-xhtml2-20060726>. (consulta 2010, 24 de abril).
- Bos, B.; Çelik, T.; Hickson, I.; Wium Lie, H. (2009). Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. W3C Candidate Recommendation.

- [en línea] Disponible en: <http://www.w3.org/TR/2009/CR-CSS2-20090908> (consulta 2010, 24 de abril).
- Bos, B.; Etemad, E. J.; Kemper, B. (2009). CSS Backgrounds and Borders Module Level 3. W3C Candidate Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2009/CR-css3-background-20091217> (consulta 2010, 24 de Abril).
- Boyer, J. M. (2009). W3C. XForms 1.1. W3C Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2009/REC-xforms-20091020/> (consulta 2010, 24 de abril).
- Brusilovsky, P.; Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. The Adaptive Web, LNCS 4321. Heidelberg, Berlin: Springer-Verlag. (3 – 53).
- Brusilovsky, P. (2001). Adaptive Hipermedia. User Modelling and User-Adapted Interaction 11. Netherlands: Kluwer Academic Publishers. (87-110).
- Bugaj, S.; Bulterman, D.; Butterfield, B.; Chang, W.; Fouquet, G.; Gran, C.; Hakkinen, M.; Hardman, L.; Hoddie, P.; Hofrichter, K.; Hoschka, P.; Jansen, J.; Kerscher, G.; Lanphier, R.; Layaïda, N.; Leif, S.; Mullender, S.; Pillet, D.; Rao, A.; Rutledge, L.; Soquet, P.; Kate, W.; Ossenbruggen, J.; Vernick, M.; Yu, L. (1998). W3C Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/1998/REC-smil-19980615> (consulta 2010, 24 de abril).
- Bulterman, D.; Jansen, J.; Cesar, P.; Mullender, S.; Hyche, E.; DeMeglio, M.; Quint, J.; Kawamura, H.; Weck, D.; García Pañeda, D.; Melendi, D.; Cruz-Lara, S.; Hanclik, M.; Zucker, D.; Michel, T. (2008). W3C. Synchronized Multimedia Integration Language (SMIL 3.0). W3C Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2008/REC-SMIL3-20081201/> (consulta 2010, 24 de abril).
- Butler, M. H. (2006). DELI: A Delivery Context Library For CC/PP and UAPerf. HP Labs. [en línea] Disponible en: <http://delicon.sourceforge.net/deliOverview.html> (consulta 2010, 24 de abril).
- Cantera Fonseca, J. M.; Lewis, R. (2009). Delivery Context Ontology. W3C Working Draft. [en línea] Disponible en: <http://www.w3.org/TR/2009/WD-dontology-20090616/> [consulta 2010, 24 de abril].
- Chisholm, W.; Vanderheiden, G.; Jacobs, I. (1999). W3C. Web Content Accessibility Guidelines 1.0. W3C Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505> (consulta 2010, 24 de abril).
- De Bra, P. (1999). Design Issues in Adaptive Web-Site Development. Proceedings of the 2nd Workshop on Adaptive Systems and User Modelling on the WWW.
- Delivery Context: Interfaces (DCI) Accessing Static and Dynamic Properties. Device Independence Working Group. [en línea] Disponible en: <http://www.w3.org/2001/di/> (consulta 2010, 24 de abril).
- Device Independence. Access to a Unified Web from Any Device in Any Context by Anyone. [en línea] Disponible en: <http://www.w3.org/2001/di/> (consulta 2010, 24 de abril).
- DIAL Part o: Primer. Device Independent Authoring Language. [en línea] Disponible en: <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/> (consulta 2010, 24 de abril).
- Fielding, R.; Irvine, U. C.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P.; Berners-Lee, T. (1999). Hypertext Transfer Protocol – HTTP/1.1. W3C – RFC2616. [en línea] Disponible en: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>. (consulta 2010, 24 de abril).

- JTelecom Italia. (2000). JADE [en línea] Disponible en: <http://jade.tilab.com/> (consulta 2010, 24 de abril).
- Kiss, C. (2007). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0. W3C Working Draft. [en línea] Disponible en: <http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430> (consulta 2010, 24 de abril).
- Klyne, G.; Reynolds, F.; Woodrow, C.; Ohto, H.; Hjelm, J.; Butler, M. H.; Tran, L. (2004). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/> (consulta 2010, 24 de abril).
- Lewis, R.; Merrick, R.; Froumentin, M. (2007). Content Selection for Device Independence (DISelect) 1.0. W3C Candidate Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2007/CR-cselection-20070725/> (consulta 2010, 24 de abril).
- Manola, F.; Miller, E. (2004). W3C. RDF Primer. W3C Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> (consulta 2010, 24 de abril).
- McGuinness, D. L.; Van Harmelen, F. (2004). OWL Web Ontology Language. W3C Recommendation. 10 February 2004. [en línea] Disponible en: <http://www.w3.org/TR/2004/REC-owl-features-20040210/> (consulta 2010, 24 de abril).
- Mérida, D.; Fabregat, R. (2002). SHAAD: sistema hipermedia adaptable, adaptativo y dinámico para la entrega de contenidos hipermedia. Proceedings of JENUI 2002. Cáceres, España.
- Mérida, D.; Baldiris, S.; Fabregat, R.; Velez, J.; Huerva, D. (2007). A User Model that incorporates characteristics of Access Devices in MAS-SHAAD. UM'07-Workshop Towards User Modelling and Adaptive Systems for All (TUMAS-A).
- Mérida, D.; Fabregat, R.; Prat, X.; Huerva, D.; Velez, J. (2008). Dynamic Content Generator for Adaptation in Hypermedia Systems. 5th. Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2008), LNCS 5149, Volumen 5149, Hannover. (320-323).
- Nielsen, H.; Leach, P.; Lawrence, S. (2000). HTTP Extension Framework. W3C RFC2774. [en línea] Disponible en: <http://www.ietf.org/rfc/rfc2774.txt> (consulta 2010, 24 de abril).
- Oppermann, R. (1994). Introduction. Adaptive User Support. En: Oppermann R. Lawrence Erlbaum. Associates, Hillsdale, New Jersey, (1-13).
- Oppermann, R.; Rashev, R.; Kinshuk. (1997). Adaptability and Adaptativity in Learning Systems. Knowledge Transfer, Volume II, Ed. A. Behrooz. (173-179).
- OWL 2 (2009). Web Ontology Language. W3C Recommendation.
- Passani, L. WURFL Open Source Project. [en línea] Disponible en: <http://wurfl.sourceforge.net/> (consulta 2010, 24 de abril).
- Smith, K. (2006). W3C Working Draft. 10 October 2006. Vodafone Group. [en línea] Disponible en: <http://www.w3.org/TR/2006/WD-dial-primer-20061010/> (consulta 2010, 24 de abril).
- Smith, K. (2007). Device Independent Authoring Language (DIAL). W3C Working Draft. Vodafone Group [en línea] Disponible en: <http://www.w3.org/TR/2007/WD-dial-20070727/> (consulta 2010, 24 de abril).
- Tarek F.; Abdelzaher, T.; Bhatti, N. (1999). Web Content Adaptation to Improve Server Overload Behavior. The 8th International World Wide Web Conference, Toronto, Ontario, Canada.
- Ubiquitous Web Applications Working

- Group. [en línea] Disponible en: <http://www.w3.org/2007/uwa/> (consulta 2010, 24 de abril).
- User Agent Profile specification. [en línea] Disponible en: <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf> (consulta 2010, 24 de abril).
- W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification.
- Waters, K.; Hosn, R. A.; Raggett, D.; Sathish, S.; Womer, M.; Froumentin, M.; Lewis, R.; Rosenblatt, K. (2007). Delivery Context: Client Interfaces (DCCI) 1.0. Accessing Static and Dynamic Delivery Context Properties. W3C Candidate Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2007/CR-DPF-20071221/> (consulta 2010, 24 de abril).
- Waters, K.; Raggett, R. A. D.; Sathish, S.; Womer, M.; Froumentin M. (2006).
- W3C Candidate Recommendation. [en línea] Disponible en: <http://www.w3.org/TR/2006/CR-DPF-20061019> (consulta 2010, 24 de abril).
- Web Accessibility Initiative (WAI). [en línea] Disponible en: <http://www.w3.org/WAI/> (consulta 2010, 24 de abril).
- Wireless Application Protocol Forum (2000). Wireless Application Protocol Wireless Markup Language Specification Version 1.3. [en línea] Disponible en: <http://www.openmobilealliance.org/tech/affiliates/wap/wap-191-wml-20000219-a.pdf>. (consulta 2010, 24 de abril).
- Wireless Application Protocol Forum (2001). Wireless Markup Language Version 2.0. [en línea] Disponible en: <http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf>. (consulta 2010, 24 de abril).

## PERFIL ACADÉMICO Y PROFESIONAL DE LOS AUTORES

**Ramón Fabregat** Ingeniero en Informática por la Universitat Autònoma de Barcelona (UAB), España. Ph.D en Ingeniería Industrial por la Universidad de Girona en 1998. Profesor del departamento de Arquitectura y Tecnología de Computadores e investigador del Instituto de Informática y Aplicaciones de la Universidad de Girona (España). Es miembro del grupo de Comunicaciones y Sistemas Distribuidos (BCDS) de la UdG y responsable del laboratorio de Sistemas Hipermedia Adaptativos.

E-mail: [ramon.fabregat@udg.edu](mailto:ramon.fabregat@udg.edu)

**David Mérida** Ingeniero Industrial y DEA en Tecnologías de la Información por la Universidad de Girona. Ha sido profesor del departamento de Arquitectura y Tecnología de Computadores durante varios años. Es miembro del grupo de Comunicaciones y Sistemas Distribuidos (BCDS) del Instituto de Informática y Aplicaciones de la UdG. Actualmente trabaja en SPAMINA como responsable del Área de Comunicaciones.

E-mail: [dmerida@eia.udg.edu](mailto:dmerida@eia.udg.edu)

**Silvia Baldiris** Ingeniera de Sistemas e Industrial de la Universidad Industrial de Santander - UIS – Colombia. Magíster en Informática Industrial y Automática de la Universidad de Girona y PhD student del programa de doctorado en Tecnología de la Universitat de Girona. Sus intereses de investigación están en los campos del modelado de usuario, la gestión del conocimiento y desarrollo de Plataformas de Educación Virtual Inteligentes.

E-mail: [baldiris@eia.udg.edu](mailto:baldiris@eia.udg.edu)

#### DIRECCIÓN DE LOS AUTORES

Grupo de Investigación BCDS  
Broadband Communications and Distributed Systems  
Universitat de Girona  
Edificio P-IV, Campus Montilivi  
17071-Girona, España

**Fecha de recepción del artículo:** 15/05/10

**Fecha de aceptación del artículo:** 17/07/10