

# CC/PP: Composite Capabilities and Preference Profiles

David Álvarez Quintana

13 de noviembre de 2006

## Resumen

Uno de los objetivos principales del W3C es el Acceso Universal. Los usuarios deben ser capaces de acceder a los contenidos con el dispositivo que prefieran, ajustándose a sus características hardware, software, infraestructura de red, lenguaje, ubicación geográfica, o habilidades físicas. CC/PP proporciona un formato estandarizado de la descripción de la información que permitirá a los dispositivos con acceso a Internet comunicar de forma efectiva sus capacidades al servidor deseado.

CC/PP es una infraestructura extensible que puede ser utilizada para comunicar el contexto de entrega desde un dispositivo al servidor Web, resultando en la entrega de contenido usable para un dispositivo dado.

## 1. Introducción

Hace unos años, la única forma cotidiana de acceder a información web era a través del PC. Sin embargo, en la actualidad el mercado ha sufrido un rápido crecimiento en lo que a dispositivos y mecanismos de acceso a la Web se refiere. Los mecanismos de conexión han evolucionado para incluir accesos de banda ancha. Al mismo tiempo, las expectativas de los usuarios en relación a acceso, disponibilidad y acceso al contenido Web, han crecido enormemente. Éste espera acceder a la información desde diferentes mecanismos de acceso, desde diferentes lugares y a diferentes momentos del día, es decir, el usuario espera tener siempre la información accesible. Ante esta demanda, los autores de contenido empiezan a desarrollar contenido accesible desde cualquier mecanismo, reduciendo esfuerzo y costes al desarrollar un único diseño de contenido susceptible de ser enviado a través de diferentes mecanismos.

### 1.1. Independencia de Dispositivo

La Independencia de Dispositivo [2] está basada principalmente en la idea de que independientemente del dispositivo usado para acceder a la información, ésta va a estar siempre disponible y accesible para el usuario, es decir, se trata de hacer la Web universal y accesible para cualquier persona, en cualquier sitio, en cualquier momento y usando cualquier dispositivo, evitando la fragmentación de la Web en espacios accesibles sólo por determinados elementos. El objetivo principal es mejorar la experiencia del usuario y al mismo tiempo reducir costes al desarrollar estándares que permitan acceder a esa información sin importar cómo.

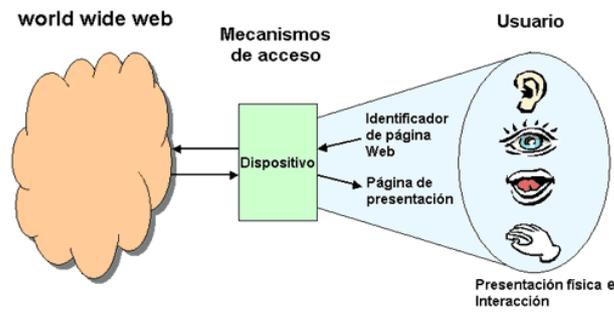


Figura 1: Desde el punto de vista del usuario

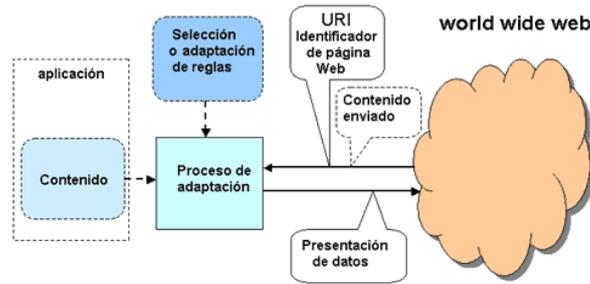


Figura 2: Desde el punto de vista del desarrollador

La Independencia de Dispositivo, desde dos puntos de vista diferentes, implicaría:

- desde el punto de vista del usuario implica acceso universal 1.1
- desde el punto de vista del programador, implica un único desarrollo válido para dispositivos con distintas capacidades 1.1

## 1.2. Preferencias de usuario

Pero no sólo debemos tener en cuenta la heterogeneidad presente entre los dispositivo de acceso al Web, sino que debemos ir más allá y pensar en los consumidores últimos del contenido Web. De la misma manera que nos encontramos ante un escenario con dispositivos cada uno de ellos con múltiples capacidades, tenemos usuarios cada uno de ellos con características culturales, físicas, etc. diferentes que deben ser tenidas en cuenta a la hora de alcanzar el objetivo de *Acceso Universal Web*.

Los usuarios deben ser capaces de ser un elemento activo en el proceso de adaptación del contenido definiendo aspectos y gustos sobre si mismos estableciendo de esta manera *condiciones* en la presentación de los contenidos. Ejemplos de éstos aspectos pueden ser el idioma nativo del usuario o su restricción de acceso a determinados contenidos.

Por tanto, debemos dotar al usuario de una infraestructura que le permita:

- Definir sus preferencias (darles un valor)
- Almacenarlas en algún tipo de repositorio de preferencias

Siendo necerio también un mecanismo que permita a los servidores de información el acceso a la definición de dichas preferencias.

## 2. El standard CC/PP

CC/PP[1] es una infraestructura que permite la representación información acerca de las capacidades del dispositivo y preferencias del usuario consumidor de contenidos por medio de *perfiles*. CC/PP se basa en RDF <sup>1</sup>, recomendación del W3C <sup>2</sup> como lenguaje de descripción de metadatos de propósito general. El estandar CC/PP define un formato para la construcción de perfiles de descripción de dispositivos y usuarios, pero no define mecanismos para el intercambio de dichos perfiles entre usuarios y servidores ni que atributos deben ser facilitados/reconocidos.

### 2.1. Estructura CC/PP

De una forma muy general, un perfil CC/PP se construye como un árbol con dos jerarquías:

- Cada perfil está compuesto de una serie de *componentes*
- Cada componente tendrá varios o al menos un atributo.

Los componentes de un perfil serán las ramas principales de un perfil CC/PP las cuales describen las características principales del cliente. Ejemplos de componentes de un perfil pueden ser la plataforma hardware, sistema operativo que corre sobre la plataforma anterior y la aplicación utilizada en el acceso.

Además, los perfiles CC/PP describen capacidades y preferencias como un conjunto de atributos para cada componente. De esta manera, la descripción de cada componente es un sub-árbol cuyas ramas serán capacidades o preferencias de dicho componente. Estos atributos podrán ser representados en ocasiones mediante valores simples, o mediante un conjunto de valores posibles como por ejemplo las versiones del protocolo HTTP soportadas por la plataforma software. Los atributos de un componente pueden ser incluidos directamente en la definición del perfil o puede ser especificado como una referencia a un perfil por defecto, el cual puede ser almacenado separadamente y accedido usando su propia URI<sup>3</sup>. El uso de perfiles externos por defecto es algo similar a la idea de herencia dinámica que posibilita importantes optimizaciones. Ya que se trata de un documento separado, puede ser almacenado y cacheado en una ubicación diferente a la del perfil concreto. Esta característica es especialmente útil en entornos *wireless* donde la transferencia de información puede ser lenta y cara. Mediante el uso de valores por defecto, sólo una pequeña cantidad de atributos deberán ser enviados desde el terminal wireless.

A continuación se muestra un perfil CC/PP completo utilizando la sintaxis propuesta por RDF. Destacar que CC/PP usa *namespaces* para distinguir vocabulario asociado con la estructura del documento (como por ejemplo *ccpp:component*) del vocabulario asociado a la propia descripción (como por ejemplo tamaño de pantalla). En el ejemplo usaremos el *namespace* *http://www.example.com/schem* asociado al prefijo *ex:* para describir todas las propiedades que no son intrínsecas a RDF ni CC/PP.

---

<sup>1</sup>Resource Description Language

<sup>2</sup>World Wide Web Consortium

<sup>3</sup>Uniform Resource Identifier

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-schema#"
        xmlns:ex="http://www.example.com/schema#">
  <rdf:Description
    rdf:about="http://www.example.com/profile#MyProfile">
    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profile#TerminalHardware">
          <rdf:type
            rdf:resource="http://www.example.com/schema#HardwarePlatform" />
        </rdf:Description>
      </ccpp:component>
      <ccpp:defaults rdf:resource="http://example.com/HWDefault"/>
      <ex:displayWidth>320</ex:displayWidth>
      <ex:displayHeight>200</ex:displayHeight>
    </ccpp:component>
    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profile#TerminalSoftware">
          <rdf:type
            rdf:resource="http://www.example.com/schema#SoftwarePlatform" />
        </rdf:Description>
      </ccpp:component>
      <ccpp:defaults rdf:resource="http://example.com/SWDefault"/>
      <ex:name>EPOC</ex:name>
      <ex:version>2.0</ex:version>
      <ex:vendor>Symbian</ex:vendor>
    </ccpp:component>
    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profile#TerminalBrowser">
          <rdf:type
            rdf:resource="http://www.example.com/schema#BrowserUA" />
        </rdf:Description>
      </ccpp:component>
      <ccpp:defaults rdf:resource="http://example.com/UADefault"/>
      <ex:name>Mozilla</ex:name>
      <ex:version>5.0</ex:version>
      <ex:vendor>Symbian</ex:vendor>
      <ex:htmlVersionsSupported>
        <rdf:Bag>
          <rdf:li>3.0</rdf:li>
          <rdf:li>4.0</rdf:li>
        </rdf:Bag>
      </ex:htmlVersionsSupported>
    </ccpp:component>
  </rdf:Description>
</rdf:RDF>

```

### 3. CC/PPex: CC/PP exchange protocol based on HTTP Extension Framework

En este capítulo presentaremos una solución al intercambio de perfiles CC/PP entre el agente de usuario y el servidor proveedor de contenidos ya que recordemos que la especificación CC/PP tan sólo define la estructura del documento que representa el perfil de usuario. Los requerimientos [4] básicos para la implementación de un protocolo de intercambio de información CC/PP son:

- El intercambio de perfiles CC/PP debe ser compatible con el protocolo HTTP/1.1 de manera que se sigan manteniendo la compatibilidad con sitios Web ya existentes.
- Debe soportar un esquema de direccionamiento indirecto con el objetivo de soportar referencias externas a perfiles.
- Los componentes utilizados en las construcciones de perfiles CC/PP deben poder ser cacheables independientemente
- Debe soportar un mecanismo que evite el reenvío de información de perfil que no haya cambiado desde la última petición que se haya realizado.

El W3C propone un protocolo de intercambio de perfiles CC/PP basado en *HTTP Extension Framework*[5]

#### 3.1. Campos de cabecera

##### 3.1.1. Profile Header

El *Profile Header* es un campo de la cabecera de la petición (request) que contendrá una lista de referencias a descripciones CC/PP. La gramática de este campo es la siguiente:

```
Profile           = profile-field-name ":" 1#reference
profile-field-name = "Profile"
reference         = <"> ( absoluteURI | profile-diff-name ) <">
profile-diff-name = profile-diff-number "-" profile-diff-digest
profile-diff-number = 1#DIGIT
profile-diff-digest = sp; < MD5 message digest encoded by base64 >
DIGIT            = <any US-ASCII digit "0".."9">
```

Cada valor del campo *reference* representa la correspondiente entidad de la descripción CC/PP. Una referencia puede ser especificada externamente mediante su URI o directamente haciendo uso para ello del campo de cabecera *profile-diff-name* que referenciará a un perfil CC/PP cuya descripción se encuentra contenida dentro de la propia petición. La capacidad de referencia a un perfil indirectamente mediante su URI posibilita una mayor eficiencia en la transacción ya que supone una considerable disminución de la cantidad de información contenida en la request. De esta manera, por ejemplo, los fabricantes de hardware podrían ofrecer las descripciones CC/PP de las capacidades de sus dispositivos en un contenedor independiente y accesible vía petición HTTP de manera que el usuario incluya en su Profile Header tan sólo una referencia a dicho perfil. El campo *profile-diff-name* referencia a un campo de la cabecera *Profile-Diff* que veremos en la siguiente sección. El campo *profile-diff-number* referencia a un determinado *Profile-Diff* ya que se posibilita que varias descripciones CC/PP viajen con el mismo nombre dentro de una misma petición. El campo *profile-diff-digest* es un resumen MD5<sup>4</sup> de la cabecera.

---

<sup>4</sup>Message-Digest Algorithm 5

### 3.1.2. Profile-Diff Header

Este campo de cabecera es el que realmente contiene la descripción CC/PP ya sea directamente o bien referenciándola mediante su URI. La gramática de este campo es la siguiente:

```
Profile-Diff           = profile-diff-field-name ":" profile-desc
profile-diff-field-name = "Profile-Diff-" profile-diff-number
profile-desc           = < the CC/PP description based on XML/RDF text format
                        (any OCTET except CTLs, but including LWS)>
```

Como podemos observar, esta cabecera se compone de un campo donde se da un nombre a la cabecera *Profile-diff*, el número con que ésta es referenciada desde la cabecera *Profile* y la propia descripción del perfil mediante el campo *profile-desc*. Podemos ver un ejemplo de extensión de cabecera a continuación:

```
Profile: "1-P1GRkSjKK50aTWXXndFcSQ=="
Profile-Diff-1: <?xml version="1.0"?>
  <RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
      xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
    <Bag>
      <Description about="HardwarePlatform">
        <Defaults>
          <Description PRF:Vendor="Nokia"
                      PRF:Model="2160"
                      PRF:Type="PDA"
                      .....
          </Defaults>
        </Description>
      <Description about="SoftwarePlatform">
        .....
    </RDF>
```

### 3.1.3. Profile-warning Header

Esta cabecera será contenida en la respuesta (response) a una petición Web y usada para transportar información acerca del resultado obtenido en el servidor una vez procesada la información de perfil contenida en la request que éste ha recibido. Un servidor debería devolver información de alerta en su respuesta si por ejemplo una de las descripciones referenciadas mediante URI en la petición no ha podido ser obtenida. La gramática de esta cabecera es la siguiente:

```
Profile-warning           = profile-warning-field-name ":" 1#warning-value
profile-warning-field-name = "Profile-Warning"
warning-value             = warn-code SP warn-target SP warn-text [SP warn-date]
warn-code                 = 3DIGIT
warn-target               = (absoluteURI | host [ ":" port ])
warn-text                 = quoted-string
warn-date                 = <"> HTTP-date <">
```

El campo *warn-code* está formado por 3 dígitos. Un código 1xx indica el estado de la descripción CC/PP (actualizada, obsoleta). Un código 2xx el tipo de adaptación que el servidor ha aplicado al contenido solicitado. El campo *warn-target* recoge la URI del host que ha producido el *warn-code*. A continuación se listan los posibles códigos de respuesta:

- **100 OK:**Esta respuesta será recibida cuando el repositorio de perfiles CC/PP proporcione información actualizada al servidor sobre el perfil requerido.
- **101 Used stale profile:**Esta respuesta será recibida cuando el repositorio de perfiles CC/PP proporcione información desactualizada al servidor sobre el perfil requerido.
- **102 Not used profile:**Esta respuesta será recibida cuando el servidor no ha podido obtener un perfil CC/PP que ha sido referenciado mediante su URI en la petición.
- **200 Not Applied:**Esta respuesta será recibida cuando el servidor no ha llevado a cabo ningún tipo de adaptación de contenido basada en el perfil CC/PP recibido.
- **201 Content generation applied:**Esta respuesta será recibida cuando el servidor ha llevado a cabo algún tipo de adaptación de contenido basada en el perfil CC/PP recibido.
- **202 Transformation applied:**Esta respuesta será recibida cuando es un proxy intermedio entre el usuario y el servidor el que ha llevado a cabo algún tipo de transformación sobre el contenido original enviado por el servidor.

Podemos ver a continuación un ejemplo de *Profile-warning Header*:

```
Profile-Warning: 101 http://www.aaa.com/hw "Used stale profile",
                102 http://www.bbb.com/sw "Not used profile",
                200 18.23.0.23:80 "Not applied" "Wed, 31 Mar 1999 08:49:37 GMT"
```

## 4. Conclusiones

Como he comentado al inicio de este trabajo, la actual situación de heterogeneidad en los dispositivos de acceso Web hace interesante el uso de un lenguaje estandarizado de descripción de recursos, como el propuesto CC/PP, con el objetivo de alcanzar el Acceso Universal.

No resulta difícil implementar una solución que enriquezca nuestro servidor web basada en el conocimiento del perfil de dispositivo. Aún así se debería contar con la adopción del estandar por parte de los fabricantes con el objetivo de facilitar los perfiles *default* de los terminales que comercialicen.

La especificación del estandar se centra casi en exclusiva en la especificación de las capacidades del dispositivo. Pero, qué ocurre con las preferencias del usuario? Recordemos que el estandar CC/PP se presenta como un lenguaje formal de descripción de ambos *recursos*. La descripción de un terminal es mucho más sencilla al poder partir de una especificación de fábrica. Pero una descripción del usuario obviamente requiere la intervención de éste. Aún partiendo de una buena especificación del usuario que se traduzca en atributos útiles para un servidor de contenidos en su proceso de adaptación queda por especificar la interfaz encargada de recoger los valores que el usuario asigne a sus preferencias. Además, deberá establecerse un mecanismo que permita recoger y actualizar estos valores en un repositorio externo de manera que se pueda hacer uso de las referencias a perfiles mediante su URI. De no ser así, el número de atributos a transmitir debería ser bajo con el objetivo de ocupar el mínimo posible de ancho de banda del canal.

Desde un punto de vista del desarrollador, el estandar no parece ser suficiente por si sola para conseguir la independencia del dispositivo. Aunque la información que arroja el estandar es un buen punto de partida, es necesario el apoyo de otras tecnologías con el objetivo de llevar a cabo un desarrollo único de contenido adaptable a todos los terminales del mercado. Actualmente, los mecanismos de adaptación de contenidos no han sido definidos formalmente, pero existe alguna tecnología ya implementada y lista para ser usada a este respecto. Como ejemplo de tecnologías a utilizar tenemos SVG para el proceso de adaptación de imagenes, o la familia de recomendaciones XSL<sup>5</sup>

Por otro lado, en este trabajo se presentó CC/PPex como protocolo de intercambio de perfiles. La presente actividad a día de hoy por parte del Grupo de Independencia de Dispositivo del W3C en una versión estandar mejorada del proceso de intercambio sugiere que CC/PPex no cubre aún todas las necesidades que deben ser exigidas. El trabajo en este punto queda por tanto abierto a nuevas soluciones.

---

<sup>5</sup>The Extensible Stylesheet Language Family

## Referencias

- [1] **Composite Capabilities/Preference Profiles Working Group**, World Wide Web Consortium <http://www.w3.org/Mobile/CCPP/>
- [2] **Device Independence Working Group**, World Wide Web Consortium <http://www.w3.org/2001/di/>
- [3] **Introduction to CC/PP technology**, IBM <http://www-128.ibm.com/developerworks/wireless/library/wi-ccpp/>
- [4] **CC/PP exchange protocol based on HTTP Extension Framework**, World Wide Web Consortium <http://www.w3.org/TR/NOTE-CCPPexchange>
- [5] **HTTP Extension Framework**, World Wide Web Consortium <http://www.w3.org/Protocols/HTTP/ietf-http-ext/draft-frystyk-http-extensions-03.txt>