

# Aplicación de las tecnologías de la Web Semántica a la problemática de cumplimentación automática de formularios en la Web Móvil \*

Diego Berrueta  
Sergio Fernández  
Ignacio Marín

Fundación CTIC de Asturias  
diego.berrueta@fundacionctic.org  
sergio.fernandez@fundacionctic.org  
ignacio.marin@fundacionctic.org

José M. Cantera  
Mario Arias  
Juan J. Hierro

Telefónica I+D  
jmcf@tid.es  
marioag@tid.es  
jhierro@tid.es

Miguel Jiménez  
Javier Soriano

Universidad Politécnica de Madrid  
mjimenez@fi.upm.es  
jsoriano@fi.upm.es

## Resumen

Una buena parte del tiempo de interacción de los usuarios con la Web se malgasta cumplimentando formularios, muchos de los cuales son muy similares entre sí. La introducción de texto libre contando con un teclado de sobremesa es una tarea ingrata. Pero en los dispositivos móviles, cuyos métodos de entrada son menos amigables, la tarea se torna inabordable. Se propone un modelo y una sintaxis para enriquecer semánticamente la definición de la interfaz de usuario con metadatos usando ontologías, con el fin de que la información correspondiente a formularios pueda en gran medida ser cumplimentada de manera automática. Se presenta un prototipo, basado en extensiones semánticas de la plataforma de software abierto MyMobileWeb, que valida las ideas expuestas en este trabajo y demuestra su viabilidad.

---

\*Trabajo parcialmente financiado por el Ministerio de Industria, Turismo y Comercio a través del Proyecto Tractor MORFEO-MyMobileWeb (FIT-350401-2006-2) y por la Consejería de Educación de la Comunidad Autónoma de Madrid y el Fondo Social Europeo mediante el plan de Formación de Personal Investigador.

## 1. Introducción

Los formularios son el método más común para la captación de datos, tanto en la web como fuera de ella. La cumplimentación de los distintos campos, sobre todo aquellos que requieren la entrada de texto libre por parte del usuario, es una labor pesada y repetitiva. Este problema es todavía más acuciante si se hace a través de un dispositivo móvil, cuyos mecanismos de entrada de información consisten fundamentalmente en teclados de doce elementos.

La situación anterior hace que la interacción del usuario con formularios en la Web Móvil esté limitada habitualmente a la utilización de listas de selección de valores. Es más, está desaconsejado y se considera una mala práctica el abuso en la utilización de campos de entrada de texto libre [1]. Sin embargo, existen aplicaciones que demandan la entrada de información textual y que se podrían beneficiar de la existencia de mecanismos de cumplimentación automática. Todo ello se une al hecho de que gran parte de la información requerida para cumplimentar los formularios puede localizarse en el propio terminal (PDA o teléfono móvil), principalmente en la agenda o algún otro formato de intercambio de metadatos, como por ejemplo vCard [2].

En este trabajo se describe un modelo formal que demuestra que el problema de la cum-

plimentación automática de formularios puede resolverse mediante la utilización de las tecnologías de la Web Semántica. La anotación semántica de los campos de un formulario permite que el agente de usuario, o un mecanismo de extensión del mismo, pueda saber el significado de cada campo y, como consecuencia de ello, rellenarlo automáticamente sin la intervención del usuario. Es más, con el fin de preservar la privacidad y la seguridad de los datos personales, el modelo considera que éstos residen y permanecen en el propio dispositivo de usuario.

En las secciones siguientes se presenta el modelo formal que resuelve el problema planteado y se describe brevemente la experimentación llevada a cabo sobre dicho modelo. Dicha experimentación se ha realizado utilizando la plataforma de software abierto MyMobileWeb, liberada en el marco la Comunidad de Software Abierto Morfeo (<http://morfeo-project.org>).

## 2. Aproximaciones actuales al problema

En la actualidad existen soluciones puramente sintácticas al problema de la cumplimentación automática de formularios, muchas de las cuales sólo se han implementando en el mundo de los navegadores de sobremesa.

El caso más típico es el navegador Mozilla Firefox, que utiliza los nombres de los campos para *autocompletar* (o presentar sugerencias), pero siempre a partir de que el usuario ya haya cumplimentado el mismo formulario al menos una vez. Al carecer de semántica, este mecanismo está imposibilitado para cubrir automáticamente distintos formularios, aunque éstos pidan al usuario los mismos datos.

También sufren limitaciones similares las técnicas de *texto predictivo*, muy extendidas en los dispositivos móviles. En este caso se presentan sugerencias procedentes de un diccionario, lo que resulta útil para reducir el número de pulsaciones al redactar mensajes, pero que no se prestan a la cumplimentación automática de formularios y no consideran ninguna información personal, que es precisamente la requerida en mayor medida por estos formularios.

Otra técnica muy utilizada son las cookies, que permiten recordar ciertos valores de campos representativos, como es el caso del login del usuario en una página que permite la autenticación. Sin embargo, esta técnica presenta muchos problemas de seguridad y es inviable e incluso ilegal su utilización para almacenar información sensible del usuario.

Estas limitaciones han llevado a que en la actualidad se considere una mala práctica el disponer de formularios en la Web Móvil con un alto contenido de campos de entrada de texto. Sin embargo, esta recomendación va en perjuicio del número de aplicaciones desplegadas en la Web Móvil y de las posibilidades ofrecidas por éstas.

El estado del arte descrito anteriormente justifica la necesidad de investigar en el desarrollo de mecanismos avanzados e inteligentes de cumplimentación de formularios, aplicables a las restricciones contextuales de la Web Móvil.

## 3. Anotación semántica de formularios

La introducción de datos en formularios Web se puede automatizar si se provee a estos últimos de una descripción que exprese, a nivel semántico, los datos que se espera recibir en cada uno de los elementos de un formulario. De esta forma no sólo el usuario es capaz de interpretar el objetivo de cada formulario, sino que su dispositivo puede detectarlo y, en caso de disponer de dicha información, introducirla en el campo correspondiente. Cabe reseñar que esta aproximación no impide que el usuario rellene con el valor que desee, sino que tan sólo es una ayuda previa, que puede ser aceptada por el usuario o corregida a voluntad.

Así mismo, la introducción de datos por parte del usuario en un formulario descrito semánticamente puede ser interpretada por el dispositivo móvil, permitiendo la ampliación de su base de conocimiento, tomando como propiedad el descriptor del formulario, y como valor aquel introducido por el usuario.

En esta sección se presenta en primer lugar un modelo formal para dotar a los formula-

rios de la semántica necesaria para permitir su cumplimentación automática. Después se profundiza en los detalles sintácticos de la relación entre el formulario y sus metadatos.

### 3.1. Modelo formal

En el caso más simple, un formulario sirve para capturar valores para un conjunto de propiedades de un sujeto. Es fácil identificar qué propiedades están presentes en un formulario, porque habitualmente hay una correspondencia explícita entre los campos que el usuario “ve” en el formulario y las propiedades. Por ejemplo, el campo “apellidos” trata obviamente de capturar el valor de la propiedad “apellidos” para un determinado individuo (sujeto). Las tecnologías para construir formularios electrónicos proporcionan abstracciones para adaptarse a los distintos tipos de propiedades (por ejemplo: campos de texto, listas desplegables, selectores de colores, etc.), que son inmediatamente reconocibles por el usuario. El sujeto, por el contrario, suele estar implícito en el formulario o en su contexto. Puede tratarse, por ejemplo, de los datos de la persona que solicita el alta en un servicio, o los datos de una cuenta bancaria a la que se realiza una transferencia. Cabe observar que el sujeto no coincide necesariamente con la persona que cumplimenta el formulario. Es más, el mismo formulario puede emplearse varias veces dentro de la misma aplicación, con distintos sujetos (por ejemplo, para captar los datos del emisor y del receptor de una transferencia).

Las tecnologías actuales para construir formularios electrónicos no contemplan la semántica de las propiedades, pero menos aún del sujeto. Desde el punto de vista formal, resolver esta deficiencia pasa por asociar explícitamente la semántica del sujeto y de las propiedades de los formularios. Afortunadamente, algunos modelos de representación del conocimiento proporcionan abstracciones muy adecuadas para realizar dicha asociación. La existencia de sujetos y propiedades forma parte, con unos nombres u otros, de formalismos como los *frames*, los modelos orientados a objetos y la lógica descriptiva [3]. Por tanto, el enriquecimiento semántico de formularios se puede llevar a

cabo asociando el sujeto y las propiedades de un formulario con los formalismos oportunos. En este caso, se propone utilizar ontologías construidas mediante lógica descriptiva.

Sea  $E$  el esquema de un formulario, determinado por una categoría de sujetos,  $S$ , y un conjunto  $F$  de campos  $f_i$  que se pretenden capturar a través del formulario. El esquema del formulario está relacionado con una categoría de sujetos (personas, cuentas bancarias, etc.), no con un sujeto particular. Los datos introducidos a través de un formulario tampoco forman parte de su esquema.

$$E = (S, \{f_1, \dots, f_n\}) \quad (1)$$

Una ontología definida mediante lógica descriptiva cuenta con un conjunto  $\mathcal{C}$  de conceptos y un conjunto  $\mathcal{R}$  de roles (propiedades).

El enriquecimiento semántico de un esquema de formulario consiste en:

- asociar  $S$  con un concepto  $C \in \mathcal{C}$ , y también
- asociar cada  $f_i$  con un rol  $r_i \in \mathcal{R}$ .

Para evitar inconsistencias, para cada  $r_i$  debe verificarse que  $\text{dominio}(r_i) \sqsubseteq C$ , es decir, que los roles son aplicables al sujeto.

Es importante destacar que esta formalización sólo cubre los casos en los cuales el formulario captura datos de un único sujeto.

En el caso más sencillo, la información que se ha de utilizar para rellenar el esquema de formulario debe estar disponible en el cliente (el navegador, el teléfono móvil, etc.). Más aún, debe estar representada utilizando un formalismo compatible con el que se ha utilizado para etiquetar el esquema de formulario, para poder trazar las correspondencias necesarias. Por tanto, si el etiquetado utiliza clases y propiedades de lógica descriptiva según una determinada ontología, la información debe tener la forma de instancias de conceptos de esa misma ontología.

Si bien las tecnologías de lógica descriptiva permiten la utilización de clases e instancias definidas según diferentes ontologías, esto requiere que estas últimas se puedan relacionar mediante técnicas de *matching* [4, 5], cuya

complejidad requiere de unos recursos computacionales que exceden los disponibles en los dispositivos móviles actuales. Es por esto que de momento no se ha abordado esta segunda aproximación.

El enriquecimiento semántico de los formularios no sólo sirve para rellenar los valores de los campos de manera automática. También puede emplearse en el sentido opuesto. Tras rellenar manualmente un formulario, el cliente tiene la opción de almacenar localmente los datos introducidos, creando una instancia de la clase  $\mathcal{C}$ , y habilitando de esta manera la cumplimentación automática en los próximos formularios similares que se encuentre el usuario.

### 3.2. Sintaxis y mecanismos de anotación

Enriquecer semánticamente un formulario requiere un lenguaje que tenga capacidades para describir los elementos que componen el formulario y mecanismos de anotación semántica de dichos elementos. A la hora de enlazar anotaciones semánticas y descripciones de la interfaz de usuario existen dos opciones:

- Separar la descripción de la interfaz de usuario de la anotación semántica, enlazando ambas mediante alguna tecnología para el establecimiento de referencias cruzadas, como es el caso de XPointer [6] para presentaciones de usuario definidas en XML.
- Extender el lenguaje de definición de interfaz con elementos adicionales que permiten al diseñador anotar semánticamente la capa de presentación al mismo tiempo que ésta es definida.

La primera opción presenta la ventaja de que no resulta intrusivo, ya que no hace falta modificar el lenguaje de descripción de interfaz. Además facilita la provisión de anotaciones semánticas por parte de un tercero, que puede no ser el creador del formulario. Sin embargo, el gran inconveniente de esta opción es que puede resultar difícil de mantener la consistencia entre la anotación semántica y la definición de interfaz, resultando en un modelo poco mantenible.

El fuerte de la segunda opción es su sencillez, unido a que permite anotar a la vez que se desarrolla la interfaz de usuario, permitiendo incluso, el desarrollo de abstracciones de anotación que eviten a los autores el conocer las tecnologías de Web Semántica (RDF [7], OWL [8], etc). Esta es la línea que persiguen tecnologías como RDFa [9] en proceso de estandarización por parte del W3C.

## 4. Experimentación sobre el modelo formal

A continuación se describe brevemente la experimentación llevada a cabo sobre el modelo formal presentado. Para ello se ha aprovechado parte de la experiencia adquirida durante la implementación una extensión de Firefox, denominada FiReX, para el auto-completado de formularios en el entorno de escritorio.

La experimentación se ha realizado sobre la plataforma de software abierto MyMobileWeb, liberada en el marco de la Comunidad de Software Abierto Morfeo (<http://morfeo-project.org>), a la que se ha provisto de soporte para anotación semántica de componentes visuales.

### 4.1. La plataforma MyMobileWeb

MyMobileWeb es una plataforma de software abierto destinada al desarrollo de sitios Web Móviles. MyMobileWeb utiliza un lenguaje declarativo, basado en XML, para la descripción estructural de la interfaz de usuario de forma independiente del dispositivo. En tiempo de ejecución MyMobileWeb se encarga de generar y adaptar dicha definición a las peculiaridades y restricciones que impone cada contexto de entrega, representado fundamentalmente por el dispositivo y navegador del usuario [10].

El lenguaje de descripción de la interfaz de usuario se basa en componentes de alto nivel estructurados en torno a contenedores que disponen dichos elementos de forma vertical, horizontal, grid, etc. MyMobileWeb proporciona distintos componentes de interfaz, entre ellos, todos aquellos que tienen que ver con la entrada de datos por parte del usuario.

En tiempo de ejecución, y en función del dispositivo de usuario, MyMobileWeb se encarga de generar el código de marcado óptimo (XHTML-MP, WML, i-Mode, etc) que implementa la especificación de interfaz definida por el diseñador. Además, MyMobileWeb se encarga de realizar automáticamente aquellas adaptaciones a múltiples dispositivos más complejas, como es el caso de paginación de contenidos extensos, transformación de imágenes, validaciones de datos, etc.

#### 4.2. Sintaxis adoptada

Para experimentar se ha elegido el Lenguaje de Definición de Presentaciones (LDP) de MyMobileWeb. Dicho lenguaje se ha extendido de forma que permite anotar semánticamente los distintos componentes de interfaz de forma directa, es decir, al mismo tiempo que se está diseñando dicha interfaz.

Los componentes visuales son anotados mediante una serie de propiedades que definen su semántica asociada y que, durante el proceso de renderización de las páginas, se convertirán en anotaciones semánticas finales en formato RDF, describiendo semánticamente los contenidos y formularios enviados al cliente. A continuación se describen los atributos utilizados para enriquecer semánticamente la capa de presentación.

- **about-resid**: define qué identificadores tendrán los conceptos de las tripletas generadas.
- **about-class**: denota la clase a la que pertenece una instancia.
- **about-prop**: define cual es el identificador de una propiedad cuyo valor es el dato representado en el componente de interfaz de usuario.
- **about-obj-datatype**: especifica el tipo de dato de un literal por medio de tipos de esquemas XML.

Además se han introducido mecanismos adicionales en el lenguaje que permiten declarar e importar las ontologías que se van a utilizar

```

<mymw:document>
<mymw:head>
...
  <mymw:import prefix="myapp"
    uri="http://www.myapp.com/"
    default="true" />
  <mymw:import prefix="foaf"
    uri="http://xmlns.com/foaf/0.1/"
    file="foaf.owl" />
</mymw:head>

<mymw:body>
<mymw:p id="p1" style="form">
  <mymw:label id="label1" for="entry1"
    style="bold">
    Name:
  </mymw:label>

  <mymw:entryfield id="entry1" style="required"
    bind="{user.name}"
    about-class="foaf:Person"
    about-prop="foaf:name" />
</mymw:p>
...
</mymw:body>
</mymw:document>

```

Figura 1: Presentación MyMobileWeb enriquecida con información semántica

para anotar semánticamente los elementos de interfaz.

La extensión del lenguaje para anotaciones semánticas permite que los diseñadores sin grandes conocimientos de RDF y OWL puedan trabajar con estas tecnologías sin mayor esfuerzo. Los componentes de tiempo de ejecución de MyMobileWeb se encargarán de generar las tripletas RDF correspondientes a la anotación semántica simplificada especificada en tiempo de diseño, incluyendo referencias a los elementos concretos del lenguaje de marcado descritos. En [11] se describe esta extensión semántica al LDP con mayor nivel de detalle.

#### 4.3. Desarrollo del experimento

Para ilustrar la solución adoptada se plantea un experimento consistente en la descripción semántica de un formulario en una presentación MyMobileWeb. Se puede ver el código fuente en la Figura 1.

Se trata de un formulario que permite introducir distintos datos personales de un individuo. Es por ello por lo que al atributo "about-class" toma el valor "foaf:person". Cada uno

de los campos del formulario representa a una propiedad de la persona, definida en la ontología FOAF. La propiedad que representa cada campo se define por medio del atributo "about-prop".

En las Figuras 2 y 3 se puede observar el RDF generado a partir de la especificación XML desarrollada. En este conjunto de tripletas se describe la clase y propiedad que se espera sea introducida en cada campo del formulario, lo que puede ser interpretado por un pequeño razonador del lado del cliente que, en caso de disponer de dicha información acerca de su usuario, es capaz de introducir el valor de dichas propiedades en los elementos correspondientes de la interfaz.

En [12] los autores describen con mayor detalle la anotación semántica de los distintos controles visuales proporcionado por el LDP.

#### 4.4. Implementación

Se ha construido un prototipo para demostrar la viabilidad de esta propuesta. El prototipo funciona sobre la plataforma MyMobileWeb y el navegador Internet Explorer Mobile, permitiendo rellenar formularios automáticamente, utilizando la información de usuario presente en la PDA (sistema operativo Windows Mobile 2005). Para ello, se han desarrollado los siguientes componentes adicionales:

- Generadores de código RDF a partir de la anotación semántica de la descripción de interfaz de usuario. Estos generadores se encargan de poner en contexto la semántica de una página cuando ésta es visitada.

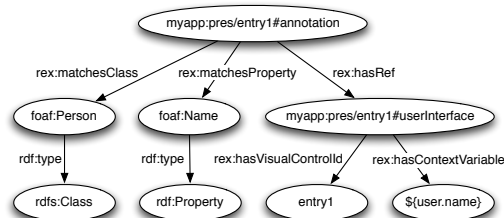


Figura 2: Grafo RDF que representa la metainformación de un control del formulario.

```

prefix myapp <http://www.myapp.com/>
prefix foaf <http://xmlns.com/foaf/0.1/>
prefix rex <http://www.fundacionctic.org/rex/1.0/>

(foaf:Person, rdf:type, rdfs:Class)
(foaf:name, rdf:type, rdf:Property)

(myapp:pres/entry1#annotation,
 rex:matchesClass, foaf:Person)

(myapp:pres/entry1#annotation,
 rex:matchesProperty, foaf:name)

(myapp:pres/entry1#annotation,
 rex:hasRef, myapp:pres/entry1#userInterface)

(myapp:pres/entry1#userInterface,
 rex:hasVisualControlId, "entry1")

(myapp:pres/entry1#userInterface,
 rex:hasContextVariable, "${user.name}")

```

Figura 3: RDF generado automáticamente a partir de los metadatos de la presentación.

- Biblioteca de cumplimentación automática de formularios en el lado cliente basada en un mini-parser RDF implementado en Javascript. Para llevar a cabo la cumplimentación automática, esta biblioteca toma como entrada el grafo RDF asociado al formulario de una página y la información de usuario.
- Implementación Javascript de la interfaz DCI [13] que permite obtener propiedades relacionadas con el dispositivo cliente. La implementación de la interfaz DCI proporciona los datos personales presentes en la PDA del usuario, todo ello a través de un componente ActiveX que permite el acceso directo a las APIs del dispositivo.
- Generador de tripletas RDF (de acuerdo a la ontología FOAF) a partir de la información de usuario presente en el dispositivo. Este componente es necesario para poder realizar correctamente el matching entre la anotación semántica y la información de usuario.

Actualmente se está trabajando en portar el mecanismo de cumplimentación automática para otros navegadores (MiniMo y Opera

Mobile), así como para dispositivos sin capacidades de scripting.

## 5. Áreas de aplicación

La tecnología propuesta, validada por medio del prototipo experimental presentado, es de aplicación a distintas problemáticas en distintos dominios, siempre desde la perspectiva del usuario en movilidad. En la mayoría de los casos presentados, los datos a cumplimentar por el usuario desde un dispositivo móvil tienen un sujeto (o sujetos) conocido que se repite a menudo en múltiples situaciones.

Las áreas de aplicación identificadas inicialmente son:

- Alta en servicios.
- Compras.
- e-Administración.
- Banca electrónica.

En todos los casos descritos anteriormente, el usuario necesita proveer, una y otra vez, los mismos datos personales, con lo cual sería muy conveniente disponer de un mecanismo automático de cumplimentación que tomara los datos directamente del dispositivo de usuario. Eso sí, en todo momento deberían establecerse los mecanismos que preserven la privacidad y seguridad del usuario.

## 6. Conclusiones y Trabajo Futuro

En este trabajo se ha demostrado que las tecnologías de la Web Semántica pueden mejorar considerablemente la experiencia del usuario en la web móvil, y como consecuencia de ello aumentar el número de aplicaciones y servicios disponibles.

Evidentemente será necesario disponer de ontologías para describir los formularios. En algunos casos estas ontologías ya existen, como en el ejemplo comentado, y en otros casos será necesario desarrollarlas. Dado que muchos formularios son esencialmente similares, existirá un alto grado de reutilización de estas ontologías.

El buen funcionamiento del prototipo hace pensar que esta técnica puede ser implementada en un futuro próximo, aunque es necesario resolver distintas cuestiones que se han dejado abiertas en el prototipo desarrollado:

- La seguridad y privacidad. Se debe alcanzar un compromiso entre la comodidad del usuario y la confidencialidad de los datos personales relativos a éste. El usuario debería ser capaz de definir sitios de confianza a los que permitiría explotar la capacidad de cumplimentación automática. A este respecto cabe destacar la plataforma P3P [14], que, hasta la fecha, no ha sido adoptada masivamente.
- En aquellas interfaces que incluyan formularios que hagan referencia a sujetos distintos pertenecientes a la misma clase, se hace necesario establecer mecanismos declarativos que permitan diferenciar entre ambos sujetos y que eviten que se cumplimenten los datos de forma incorrecta.
- Es necesario extender el modelo de anotación semántica para aquellos casos en los que un formulario haga referencia a varios sujetos pertenecientes a clases distintas. Una vez extendido el modelo, se especificará la sintaxis que lo dé soporte.
- Cuando no se disponga a priori de información para llevar a cabo la cumplimentación automática, es necesario desarrollar elementos software que permitan “recordar y almacenar” localmente en el dispositivo los datos introducidos inicialmente en los formularios por parte del usuario. Dichos datos serán utilizados para la cumplimentación automática en sucesivas apariciones del mismo formulario o de otro formulario que semánticamente equivalente. Esta funcionalidad está disponible en FiReX y está en fase de diseño su portado al entorno móvil.
- Es necesario que el componente de cumplimentación automática tenga en cuenta

las posibles equivalencias entre la/s ontología/s con la que se anotan los formularios y la/s ontología/s en las que los datos están representados.

- Se deben tratar aquellos casos en los que el proveedor de los metadatos no sea necesariamente el proveedor del formulario, pudiéndose de esta manera acelerar la implantación de los servicios clave. A este respecto se están adoptando parte de las ideas ya existentes en la extensión FiReX para Firefox.

Finalmente cabe mencionar que, dado que esta propuesta surge para facilitar la utilización de formularios, y por tanto incrementa la usabilidad de los servicios sin incurrir en un incremento de su coste, es previsible que logre aceptación tanto por los proveedores de los servicios como por los consumidores.

## Referencias

- [1] mobile Top Level Domain (mTLD): dot-Mobi mobile web developers guide. Technical report, mTLD (2006)
- [2] Iannella, R.: Representing vCard objects in rdf/xml. Technical report, W3C (2001)
- [3] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
- [4] Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. **3730** (2005) 146–171
- [5] Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. *Knowl. Eng. Rev.* **18** (2003) 1–31
- [6] DeRose, S., Jr., R.D., Grosso, P., Maler, E., Marsh, J., Walsh, N.: XML pointer language (XPointer). Technical report, W3C (2002)
- [7] Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): Concepts and abstract syntax. Technical report, W3C (2004)
- [8] Dean, M., Schreiber, G.: OWL web ontology language reference. Technical report, W3C (2004)
- [9] Adida, B., Birbeck, M.: RDFa primer 1.0. Technical report, W3C (2006)
- [10] Soriano, J., Jimenez, M., Cantera, J.M., Hierro, J.J.: Delivering mobile enterprise services on morfeo's mc open source platform. *Mobile Data Management, IEEE Computer Society* (2006) 139–146
- [11] Soriano, J., Lopez, G., Jimenez, M., Fernandez, R., Hierro, J.J.: Semantic web content adaptation and services delivery on morfeo's semantic mobility channel. *Mobile Data Management, IEEE Computer Society* (2006) 78–85
- [12] Alonso, F., Frutos, S., Jiménez, M., Soriano, J.: Semantic repurposing and personalization of web contents in ubiquitous and mobile computing environments. In et al, T.P., ed.: 2nd Int. Workshop on Semantic Web Technologies for Ubiquitous and Mobile Applications (SWUMA 2006), ECAI 2006, Trentino, Italy, August 28-September 1, 2006, Proceedings, ECCAI (2006)
- [13] Waters, K., Hosn, R.A., Raggett, D., Sathish, S., Womer, M., Froumentin, M.: Delivery Context Interfaces (DCI): Accessing static and dynamic properties. Technical report, W3C (2006)
- [14] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J.: The Platform for Privacy Preferences 1.0 (P3P1.0) specification. Technical report, W3C (2002)