Integración de tecnología webcrawler en sistemas de gestión de fuentes de información: Desarrollo de la aplicación Cumulus2

Manuel BLÁZQUEZ OCHANDO
Dpto. de Biblioteconomía y Documentación, Facultad de Ciencias de la Documentación de la UCM
C/Santísima Trinidad, 37. CP 28010 Madrid, España

Esmeralda SERRANO MASCARAQUE
Dpto. de Ciencias de Sanitarias y Medicosociales, Facultad de Documentación de la UAH
C/San Cirilo s/n. CP 28805 Madrid, España

RESUMEN

El objetivo de la investigación es elaborar una herramienta especializada en la gestión de fuentes de información en ciencia y tecnología que, haciendo uso de las técnicas de de análisis parser, sea capaz de mejorar las capacidades de recuperación de contenidos en la unidad o centro de información y documentación; así como solucionar el actual problema que supone el alto nivel de obsolescencia de la literatura científica. En esta línea se ha tomado como base el programa Cumulus, al cual le ha sido integrado un programa webcrawler que actuará en fase de pre-catalogación, recuperando la mayor cantidad de información posible, correctamente identificada. Para ello, se presentan pruebas metodológicas y cuantitativas que permiten contrastar y repetir los resultados obtenidos. Finalmente, como resultado de la gestión y edición semi-automática de las fuentes de información, se presenta un modelo de directorio web para la correcta representación y visualización de los contenidos, atendiendo a los principios de usabilidad y accesibilidad web.

Palabras clave: Fuentes de información, ciencia y tecnología, recuperación de información, usabilidad y accesibilidad web, webcrawler, automatización, herramientas bibliográficas, cibermetría, PHP DOM.

1. INTRODUCCIÓN

El problema de la obsolescencia de las fuentes de información científica [1] constituye una constante comúnmente estudiada en los análisis bibliométricos y posteriormente cibermétricos, con el objetivo de desvelar la vida útil de las publicaciones científicas. Esta situación implica que cualquier sistema de información que catalogue documentación científica, al poco tiempo de ponerse a disposición de la comunidad científica, se volverá obsoleta y será superada por la novedad de cualquier otro descubrimiento o ensayo más reciente.

Una posible solución al problema de la obsolescencia es el empleo de programas webcrawler especializados en el rastreo y recuperación constante de los contenidos que haga efectivo un seguimiento automático de la evolución de una fuente de información. En esta línea se enmarca el objeto de estudio que se plantea, el desarrollo de un sistema que integre la tecnología de webcrawling y los métodos de catalogación especializados en recursos y fuentes científicas.

2. INTEGRACIÓN DE TECNOLOGÍA WEBCRAWLER

Utilizando como punto de partida el programa Cumulus1 [2], se desarrollará una segunda versión cuyas principales mejoras residen en la integración de un módulo webcrawler capaz de recopilar información sobre los recursos o fuentes que se estén catalogando en el sistema.

Para la consecución de este objetivo, se llevan a cabo los siguientes pasos; a) diseño del flujo de trabajo de la herramienta, b) modelo de webcrawler, c) proceso de integración, d) prueba de funcionamiento y e) visualización y representación.

a) **Diseño del flujo de trabajo:** Se concibe una fase de pre-catalogación en la que el usuario registra la URL de la fuente de información.

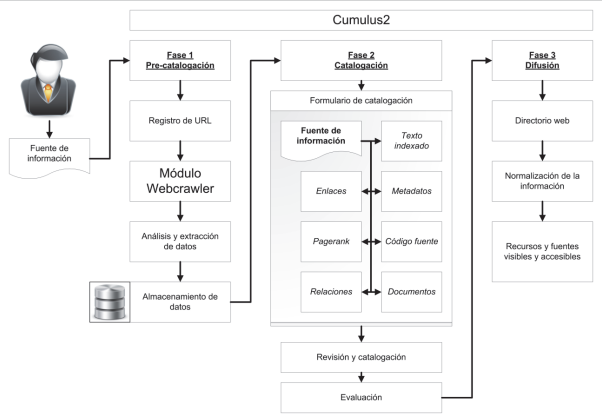

Figura 1. Flujo de trabajo de Cumulus2. Véase diagrama completo disponible en: http://www.mblazquez.es/documents/cumulus2overview.png

Dicha dirección es cargada por el módulo de webcrawler que analiza y extrae la información de la fuente o recurso que se pretende describir. Los contenidos obtenidos (metadatos, enlaces, documentos, texto completo y código fuente) son almacenados en base de datos para completar la ficha descriptiva de la fuente.

Ésta será supervisada y completada por el usuario de tal forma que el proceso de análisis es asistido automáticamente por el sistema. Finalmente, se prepara la información para su difusión, generando un directorio con todos los registros de fuentes y recursos descritos, aplicando métodos de usabilidad y accesibilidad web. *Véase figura1*.

b) Modelo de webcrawler: El programa webcrawler ha sido programado en lenguaje PHP a partir de la librería cURL [3], para facilitar su integración con el programa Cumulus2.

Su funcionamiento estipula la recepción de una dirección URL que es depurada de espacios y caracteres especiales. A continuación, se comprueba su disponibilidad en red, efectuando para ello una petición vía socket que evita fallos de conexión y finalización prematura del proceso, *véase figura2*.

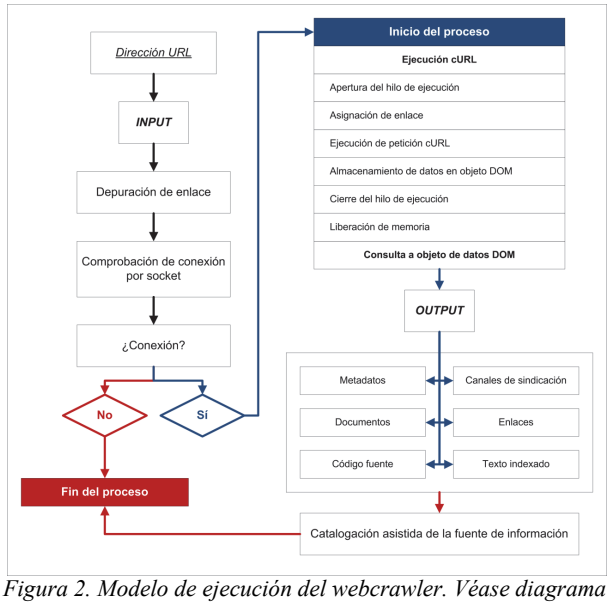

Figura 2. Modelo de ejecución del webcrawler. Véase diagrama completo disponible en: http://www.mblazquez/documents/cumulus2crawlerprocess.png

Si se recibe eco de respuesta, se desencadena un proceso que comprende la apertura de un hilo de ejecución para dicha URL, obteniendo como resultado el código fuente de la página web objetivo, que es almacenado en un objeto DOM [4], *véase tabla1*.

```
$thread1 = curl_init();
curl_setopt($thread1, CURLOPT_URL, $url1);
curl_setopt($thread1, CURLOPT_USERAGENT,
$cf_agent);

curl_setopt($thread1, CURLOPT_HTTPHEADER,
array($cf_header));

curl_setopt($thread1, CURLOPT_FAILONERROR, true);
curl_setopt($thread1, CURLOPT_FOLLOWLOCATION,
true);

curl_setopt($thread1, CURLOPT_LOW_SPEED_TIME, 3);
curl_setopt($thread1, CURLOPT_LOW_SPEED_LIMIT,
1048576);

curl_setopt($thread1, CURLOPT_AUTOREFERER, true);
curl_setopt($thread1, CURLOPT_RETURNTRANSFER,
true);

curl_setopt($thread1, CURLOPT_FORBID_REUSE, true);
curl_setopt($thread1, CURLOPT_FRESH_CONNECT,
true);

curl_setopt($thread1, CURLOPT_BUFFERSIZE,
$cf_buffer);

curl_setopt($thread1,
CURLOPT_DNS_CACHE_TIMEOUT, $cf_timecache);
```

```
curl_setopt($thread1,
CURLOPT_CONNECTTIMEOUT_MS, $cf_timeconnect);
curl_setopt($thread1, CURLOPT_TIMEOUT_MS,
$cf_timeout);

$html1 = curl_exec($thread1);
curl_close($thread1);

$dom1 = new DOMDocument();
@$dom1->loadHTML($html1);
$xpath1 = new DOMXPath($dom1);
```

Tabla 1. Apertura de hilo de ejecución y almacenamiento DOM

Antes de la ejecución cURL, se establecen una serie de parámetros que ayudan a controlar su funcionamiento. Los más importantes son el nombre del agente webcrawler que efectúa la operación, la cabecera HTTP de la petición cURL en la que se establecen las extensiones MIME de los contenidos aceptados, el tamaño de buffer, el tiempo que se mantiene la información en cache, el número de milisegundos de espera mientras el sistema se conecta al recurso y el tiempo de ejecución de cURL en milisegundos.

Todos estos parámetros han sido integrados en el área de configuración de la herramienta Cumulus2 de forma tal, que puedan ser adaptados para su refinamiento. Para recuperar la información contenida en el objeto DOM se emplea una técnica de consulta basada en XPath [5], consistente en señalar la ruta de acceso al elemento deseado, obteniendo rutinas similares a las mostradas en la *tabla2*.

```
if($cf_metadata == 'on'){
      $varMETAS1 = $xpath1->query("/html/head//meta");
      for($i=0; $i<$varMETAS1->length; $i++) {

$itemMETAS1 = $varMETAS1->item($i);
$arrMETAS1[] = $itemMETAS1->getAttribute('content');

      }
      $metas1 = @implode("|",$arrMETAS1);
} else { $metas1 = ""; }
```

Tabla 2. Rutina de recuperación de metadatos

No obstante este proceso de recuperación de información, previo análisis de estructuras y etiquetas, implica procesos de tratamiento y depuración de los contenidos [6]. Esto es la verificación de los enlaces obtenidos, la eliminación de códigos y etiquetas para la extracción del texto completo, la eliminación de valores duplicados, de dobles espacios, saltos de línea, marcas de párrafo, la conversión a set de caracteres normalizado de los textos obtenidos y la comprobación de las extensiones de los documentos enlazados en cada hipervínculo.

Adaptando el código anteriormente expuesto y efectuando la depuración pertinente mediante el uso de expresiones regulares, se obtienen los siguientes contenidos, *véase tabla3*.

Metadatos	Tanto metadatos como meta-etiquetas pueden ser distinguidos según tipo de etiqueta ya predefinido.
Meta-etiquetas	
Canales de sindicación	Los canales de sindicación son obtenidos a partir de las etiquetas link y su tipo MIME. Por ejemplo: application/rss+xml
Código fuente completo	Se extrae directamente como resultado de la ejecución cURL
Texto completo	Se eliminan todos los contenidos javascript del código fuente y a continuación todas las etiquetas HTML, saltos de línea, tabulaciones duplicadas, saltos de página y etiquetas especiales, quedando únicamente el texto del recurso listo para su indexación.
Mapa de enlaces	Todas las direcciones o enlaces URL presentes en el código fuente son recopiladas, generando un mapa de enlaces del recurso. Posteriormente son analizados para eliminar enlaces javascript y enlaces vacíos de tipo #. A continuación, se transforman todos los enlaces relativos a notación absoluta, eliminándose duplicados y obteniendo como resultado un conjunto ordenado y normalizado.
Imágenes	Las imágenes son obtenidas al filtrar el mapa de enlaces anterior por las extensiones de archivo más frecuentes. Por otro lado, se suman todos los enlaces obtenidos a través del atributo *src* de la etiqueta de imagen <*img*>. Al igual que en el caso anterior, éstas son depuradas y normalizadas, eliminando duplicados.

Documentos	Se obtienen al filtrar el mapa de enlaces por las extensiones de tipo de documento más comunes.
Títulos, titulares y párrafos	Tanto los títulos como titulares son recuperados a partir de las etiquetas HTML destinadas a cada fin. En cambio, los párrafos exigen un esfuerzo de interpretación superior ya que varían de un diseño web a otro. Por ejemplo, el empleo de etiquetas <*div*>, <*span*>, <*p*> o <*blockquote*>.

Tabla 3. Contenidos recuperados con el webcrawler

c) **Proceso de integración:** La integración del webcrawler requiere establecer un mapa de flujo de datos con el que se canaliza la entrada y almacenamiento de información en la base de datos común. Tal como se muestra en la *figura3*, algunos de los campos de descripción más significativos de una fuente de información pueden ser descritos mediante este proceso de forma automática, tales como el título, el resumen, el mapa de contenidos o las autoridades.

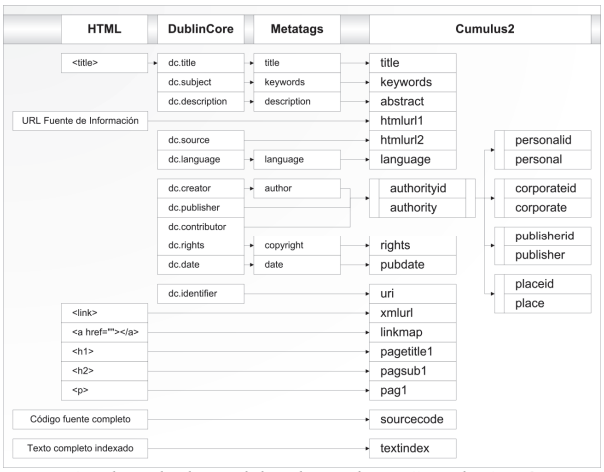

Figura 3. Flujo de datos del webcrawler a Cumulus2. Véase diagrama completo disponible en: http://www.mblazquez.es/documents/cumulus2dataflow.png

Aplicando lo anteriormente enunciado, de los 85 campos que componen la tabla principal del programa Cumulus2, unos 50 están destinados a la descripción del recurso; de los cuales 30 pueden ser completados automáticamente.

Esto supone que en el mejor de los casos el 60% de los campos de análisis estarán rellenos. Ello dependerá de múltiples factores pero en esencia, del código fuente de cada recurso, su meta-descripción, y en definitiva de la riqueza y extensión de sus contenidos.

d) **Prueba de funcionamiento:** Consiste en ejecutar de forma independiente el webcrawler con varias fuentes de información de reconocido prestigio internacional. El objetivo de la prueba es obtener la mayor cantidad de información posible de su primera página de portada.

Este método desvela, tal como se muestra en la *tabla4*, el tipo de información que es capaz de reconocer y la cantidad de datos o contenidos que puede recuperar.

	f1	f2	f3	f4	f5
Metadatos DC	0	0	4	0	0
Meta-etiquetas	2	2	4	0	6
Enlaces	130	114	49	45	95
Documentos	0	0	0	1	2
Imágenes	22	44	21	10	27
Multimedia	0	0	0	0	0
Canales sindicación	0	0	1	1	8
Títulos	1	1	1	1	1
Titulares	33	4	0	15	65
Párrafos	26	4	2	4	26
Código fuente (caracteres)	111.971	54.026	17.632	17.921	40.821
Texto indexado (palabras)	689	361	272	519	1.368
Tiempo de carga (microseg)	0,73	1,12	0,82	1,38	1,09

– f1. CSIC Consejo Superior de Investigaciones. Científicas. *Disponible en: http://www.csic.es/* – f2. NIH National Institutes of Health. *Disponible en: http://www.nih.gov/* – f3. NOAA National Oceanic and Atmospheric Administration. *Disponible en: http://www.noaa.gov/* – f4. MPG Max Planck Gesellschaft. *Disponible en: http://www.mpg.de/en* – f5. IAEA International Atomic Energy Agency. *Disponible en: http://www.iaea.org/*
Esta prueba ha sido diseñada para ser repetida y contrastada con los mismos recursos o cualquier otro que el lector estime de interés. Consúltese: – BLÁZQUEZ OCHANDO, M. Primeras pruebas del mbot webcrawler. Disponible en: http://www.mblazquez.es/documents/articulo-pruebas1-mbot.html

Tabla 4. Resultados de la prueba de funcionamiento

e) Visualización y representación de resultados en directorio: Al igual que en versiones precedentes, Cumulus2 consta de un directorio que permite aglutinar y recuperar todas las fuentes de información y recursos que fueron catalogados. La principal novedad viene dada por un nuevo modelo de accesibilidad y usabilidad de los contenidos, que ha sido desarrollado ex profeso para tal caso, denominado SRW Schematic Reduction of Websources [7]. Esta técnica tiene como objetivo utilizar el mínimo número de etiquetas HTML para representar los contenidos que fueron objeto de análisis para el webcrawler. Dicho de otra forma, transformar un recurso original en su mínimo denominador común sin perder su significación original [8].

El resultado de aplicar este proceso desemboca en la eliminación de todas las imágenes originales, interfaz gráfico, códigos javascript y tablas, en beneficio del empleo de encabezados, lista de enlaces a contenidos y extractos textuales correspondientes a las principales áreas de descripción. *Véase figura4*.

Figura 4. Recurso del CSIC y su ficha SRW

3. CONCLUSIONES

1. La integración de programas webcrawler, en sistemas de gestión de fuentes de información, puede reducir la carga de trabajo del documentalista considerablemente. En el caso del programa Cumulus, hasta en un 60% cuando se procede al análisis exhaustivo de los recursos.

2. Cualquier proceso de integración similar requiere un método que mida la capacidad de extracción de datos del webcrawler sobre un recurso; calculando la eficiencia a partir del número de contenidos recuperados en relación al número total de contenidos disponibles. Por otro lado, resulta necesaria la planificación del flujo de datos hacia los campos que conforman la base de datos, común con el sistema de análisis o descripción.

3. El desarrollo de funciones de crawling basadas en PHP DOM facilita en gran medida el análisis de las estructuras y contenidos de cualquier recurso web, siendo posible la recuperación mediante consultas XPath. No obstante, se encuentran dificultades que atañen al reconocimiento de párrafos y bloques de contenidos, debido a la falta de normalización en la delimitación de los recursos de tales objetos.

4. El programa Cumulus2 es capaz de recopilar la información básica de cada fuente de información y representarla mediante un directorio accesible y simple; de tal forma que, recursos *a priori* inaccesibles puedan ser transformados y consultados en su máxima reducción esquemática sin perder la significación de sus contenidos.

4. BIBLIOGRAFÍA

[1] BURTON, R. E. y R. W. KEBLER. 1960. The Half-Life of some Scientific and Technical Literatures. *American Documentation*. 11, pp.18-22.

[2] BLÁZQUEZ OCHANDO, M. 2010. *Gestión de fuentes de información en ciencia y tecnología: desarrollo del programa CUMULUS. En:* VII Seminario Hispano-Mexicano de Biblioteconomía y Documentación. México DF: CUIB.

[3] *Client URL Library*. 2011. [online]. [Consultado: 19 Feb 2011]. Disponible en: http://php.net/manual/es/book.curl.php

[4] *Document Object Model*. 2011. [online]. [Consultado: 19 Feb 2011]. Disponible en: http://php.net/manual/es/book.dom.php

[5] *SimpleXMLElement class*. 2011. [online]. [Consultado: 19 Feb 2011]. Disponible en: http://www.php.net/manual/en/class.simplexmlelement.php

[6] LI, Y. y J. YANG. 2009. A novel method to extract informative blocks from web pages. *En*: *International Joint Conference on Artificial Intelligence (JCAI)*. Haikou, pp.536-539.

[7] BLÁZQUEZ OCHANDO, M. y E. SERRANO MASCARAQUE. 2011. *SRW Schematic Reduction Website*. [online]. [Consultado: 19 Feb 2011]. Disponible en: http://www.mblazquez.es/documents/articulo-tecnica-srw.html

[8] BOK KIM, Y. 2010. Accessibility and Usability of User-centric Web Interaction with a Unified-Ubiquitous Name-based Directory Service. *Journal World Wide Web*. 13(1-2), pp.107-108.