

Streams basados en RSL para métricas de Posicionamiento Web

C. Salgado, M. Peralta, D. Riesco, G. Montejano
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Capital, Argentina
Ejército de los Andes 950 – San Luis – San Luis – Argentina
C.P.: 5700
Tel.: 54-02652-424027 – Int. 251
[csalgado, mperalta]@unsl.edu.ar

1. Introducción

El constante crecimiento de la Web, y la cantidad de sitios de temáticas similares, ha llevado a una amplia difusión y utilización de los motores de búsqueda, lo que hace indispensable para el éxito o fracaso de un sitio Web poder aparecer entre las primeras páginas de los buscadores para así obtener un mayor número de visitas. Esto ha llevado a la búsqueda y creación de métodos y técnicas que permitan posicionar mejor una página o sitio web entre las primeras que aparecen en los resultados mostrados por los buscadores. La aplicación de estas técnicas es lo que se denomina Posicionamiento Web.

En este sentido, nuestro trabajo se enfoca en la obtención de parámetros que permitan medir y establecer cuán bien posicionado se encuentra un sitio web. Para ellos estamos definiendo un conjunto de métricas que nos puedan dar indicadores a tal fin. Como es sabido, toda métrica para que sea útil y aplicable debe estar definida de manera confiable y formal. Por ello consideramos que el formalismo de *Streams* [1] es una herramienta que puede ser aplicada para la especificación formal de este tipo de métricas, en conjunto con la aplicación de RAISE para especificar las distintas componentes streams [2] con la finalidad de establecer chequeos automáticos, no provistos por los streams, con las herramientas del método RAISE [3].

En este artículo presentamos la especificación, usando el lenguaje RSL, de la definición basada en el formalismo de streams [4], de una métrica web que cuenta las ocurrencias de un término dado como elemento de búsqueda en un sitio web.

En las secciones siguientes presentamos una breve introducción a los conceptos de posicionamiento Web, Streams, RAISE y RSL, como así también se describe cómo se especifica una métrica web utilizando las especificaciones RSL de los transformadores de streams .

2. Posicionamiento Web

Después de realizar una gran inversión para desarrollar un sitio Web, las empresas descubren que el mismo a menudo resulta invisible para los usuarios, ya que no aparecen en los resultados de búsqueda. **El no aparecer en las primeras posiciones de la búsqueda y no estar visible en Internet es una desventaja competitiva que las empresas buscan solucionar. Resulta muy poco productivo tener un sitio muy funcional y bien diseñado gráficamente, si los usuarios no lo encuentran en Internet.**

La mayoría de los usuarios de Internet utilizan los motores de búsqueda para encontrar la información o productos que necesitan, por lo tanto es imprescindible que encuentren su web en las primeras posiciones. Para lograrlo, es necesario desarrollar y planificar una buena estrategia que permita ubicar el sitio entre las primeras posiciones de los resultados obtenidos en las búsquedas.

En este sentido es que surge lo que se denomina *Posicionamiento Web* (PageRank) que es *el proceso de intentar maximizar la exposición de un sitio en varios motores de búsqueda y directorios, mediante la utilización de palabras claves y frases específicas. La mecánica consiste principalmente en realizar cambios al sitio (título de la página, desarrollo de un contenido rico en*

palabras clave importantes, utilización de META-datos) con la finalidad de hacerlo más atractivo para los motores de búsqueda [5]. Como resultado de su aplicación se logra una mejor ubicación de un sitio en los buscadores.

El posicionamiento y la optimización de un sitio van ligados a los esfuerzos para lograr una posición destacada en el resultado de una búsqueda realizada sobre algún tema de interés.

El posicionamiento en Internet requiere que los sitios web estén bien diseñados y sean fácilmente navegables. Por ello consideramos de mucho interés tener herramientas que permitan medir y mejorar la calidad de un sitio web de manera que los motores de búsqueda los posicionen dentro de los primeros. En este sentido nuestro trabajo apunta a proveer una herramienta que ayude a los diseñadores a crear sitios que estén bien posicionados en la web.

3. Streams

Un stream es una secuencia finita de mensajes transmitidos sobre un canal de comunicación. Dado un alfabeto A , el conjunto de streams finitos A^* está formado por todas las n -uplas $A = \langle a_1, \dots, a_n \rangle$ de longitud $|A| \geq 0$, con cada elemento $a_i \in A$.

Entre las operaciones primitivas definidas por el formalismo de streams se encuentran: Añadir un elemento al frente del stream; dicha operación se denota por \blacktriangleleft y su funcionamiento es el siguiente: $a_0 \blacktriangleleft \langle a_1, a_2, \dots, a_n \rangle = \langle a_0, a_1, a_2, \dots, a_n \rangle$. Cabe destacar que todo stream se genera, a partir del stream vacío denotado por $\langle \rangle$, usando esta operación. La concatenación de dos streams, denotada por $\&$, trabaja como sigue: dado dos streams $\langle a_0, a_1, \dots, a_n \rangle$ y $\langle b_0, b_1, \dots, b_m \rangle$, produce el stream $A \& B = \langle a_0, a_1, \dots, a_n, b_0, b_1, \dots, b_m \rangle$.

La potencia en la definición de componentes de los streams, se ve incrementada por el uso de otra de sus construcciones primitivas: los transformadores de streams o equivalentemente funciones de procesamiento de streams. Estas funciones mapean streams de entradas en streams de salida y pueden modelar el comportamiento de componentes determinísticas de n entradas y m salidas. Simbólicamente $F: A_1^* \times \dots \times A_n^* \rightarrow B_1^* \times \dots \times B_m^*$. Con estos constructores primitivos se han definido un conjunto de estándares funcionales de streams que son de amplia utilización en la especificación de componentes de software. Por ejemplo, $\text{map}_n(f)$: aplica una función, que recibe como parámetro, a todos los elementos del stream de entrada; $\text{scan}(s, \oplus)$: aplica la función binaria \oplus a todos los elementos del stream de entrada; zip une pares de streams en un stream de pares, etc. Una descripción detallada de cada uno de estos transformadores de stream puede ser vista en [1].

El formalismo de los streams permite modelar componentes de software. Si bien, existen diferentes métodos para lograr este objetivo, los streams presentan la ventaja de hacer énfasis en el comportamiento de las entradas y salidas de las componentes de software especificadas. Esta característica presenta una perspectiva diferente y de valor en el ámbito de los métodos de especificación. Es importante destacar que el uso del formalismo de streams en la especificación de componentes de software promueve la reutilización de componentes, lo cual es una característica importante en el diseño y construcción de software.

4. El Método Formal RAISE

Hay dos actividades principales en el método RAISE: Escribir una especificación inicial y desarrollarla hasta algo que puede ser implementado en un lenguaje de programación. La especificación inicial es la tarea más crítica porque si se parte de una especificación errónea no sólo el sistema no cumplirá con los requerimientos del usuario, sino que además se gastará mucho esfuerzo en la construcción de algo incorrecto.

Los problemas de escribir especificaciones incorrectas surgen generalmente porque se parte de descripciones informales de los requerimientos del mismo, ya que los expertos en el dominio son usuarios comunes, y en sus explicaciones tienden a perder detalles que son de mucha importancia para la construcción de la especificación inicial.

Las especificaciones de los sistemas se construyen generalmente con lenguajes informales, que si bien brindan una visión que ayuda a una comprensión más estrecha del sistema de estudio, carecen de métodos para corroborar que la misma es correcta.

Esto se puede ver en casos más sencillos de especificaciones, como por ejemplo, la de un algoritmo. Usualmente para realizar esta tarea se utiliza pseudo código o lenguaje natural, éstas herramientas informales, son ampliamente utilizadas y queda a criterio de quién la escribió la evaluación de su corrección. Esta actividad dificulta la tarea del programador ya que no solo debe crear el algoritmo sino que además debe estar seguro de que lo que escribió es lo que el algoritmo realmente debe realizar.

Según el método RAISE, una especificación debe:

- Capturar los requerimientos de una manera precisa, no ambigua.
- Contener solo lo esencial del sistema.
- Ser escrita con los conceptos que el usuario maneja.
- Ser clara y se debe concentrar en aquellas cosas que son problemáticas.
- Minimizar los estados.
- Identificar condiciones de consistencia.

Si se parte de una especificación inicial que cumpla estos criterios se pueden ir derivando especificaciones más detalladas del sistema que se desea construir, en éste punto el método RAISE permite verificar que la segunda especificación sea un refinamiento de la primera.

En la siguiente sección presentamos la aplicación de estos formalismos para la definición y especificación de una métrica web que permite la valoración del posicionamiento web en los motores de búsqueda de los buscadores de un sitio web.

5. Especificación en RSL de la Métrica NTERM: Número de apariciones del término motivo de Búsqueda

Basándonos en la definición de métrica propuesta por [6] que dice que una métrica es la correspondencia de un dominio empírico (mundo real) a un mundo formal, matemático, hemos definido un conjunto de métricas que nos permitirán evaluar el posicionamiento de un sitio web. De dichas métricas presentamos en el presente trabajo *NTERM*, la cual cuenta el número de apariciones de un término motivo de búsqueda en un sitio determinado

NTERM (t,s)= Número de apariciones del término motivo de búsqueda *t* en el sitio web *s*.

Cabe destacar que el termino motivo de búsqueda para esta primer instancia consta de solo una palabra. Como próximo paso se definirá una métrica que permita analizar términos más complejos.

Para la especificación de la métrica hemos definido dos transformadores stream: *EncontrarTermino* y *Sumador*.

El transformador *EncontrarTermino* recibe como entrada un stream de streams de código html correspondientes a las páginas a analizar más el término motivo de búsqueda. Por cada página encuentra las ocurrencias del término e incorpora un stream al stream de salida indicando las ocurrencias del mismo.

El transformador *EncontrarTermino* recibe como entrada un stream de streams de código html correspondientes a las páginas a analizar más el término motivo de búsqueda. Por cada página encuentra las ocurrencias del término e incorpora un stream al stream de salida.

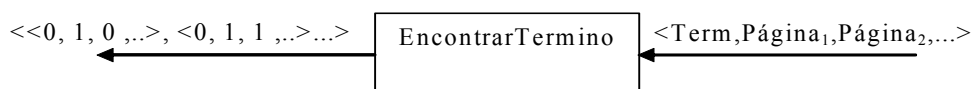


Figura 1: Visión de caja negra del transformador de stream *EncontrarTermino*

El dominio de EncontrarTermino está formado por el término motivo de la búsqueda y las páginas pertenecientes al sitio analizado. Éstas pueden expresarse como sigue:

- Palabras Claves = Palabras claves del lenguaje html.
- Identificadores = Texto definido por el usuario dentro de la página.
- Término = Palabra definida para el usuario a ser buscada.
- Página = (Palabras Claves \cup Identificadores)*
- Páginas = Página*

El rango de EncontrarTermino está formado por streams de streams de 0 y 1, donde 0 indica que el término encontrado no es el buscado y 1 indica que si lo es.

Este transformador, hace uso de la función auxiliar *Busqueda*. La cual encuentra las ocurrencias del término motivo de la búsqueda en una página.

Una vez establecido el dominio y rango del transformador se puede especificar su comportamiento en RSL [7] como sigue:

```

schema EncontrarTermino =
class
  Term, Pagina, Sitio = Pagina-list

value
  busqueda: Term x Pagina  $\rightarrow$  Int-list
  busqueda (t,p) is
    if p= $\diamond$  then  $\diamond$ 
      elseif hd(p)=t then add (1,busqueda(t,tl(p)))
      else add (0,busqueda(t,tl(p)))
    end,

  busqSitio: Term x Sitio  $\rightarrow$  (Int-list)-list;
  busqSitio (t,s) is
    if (s= $\diamond$ ) then  $\diamond$ 
      else add(busqueda(t,hd(s)),busqSitio(t,tl(s)))
    end.

Especificación 1: Transformador de streams EncontrarTermino
  
```

El transformador de streams Sumador, recibe como entrada un stream de streams de 0 y 1 producto de la salida del transformador EncontrarTermino y su rango son los enteros, indicando el número de ocurrencias en el sitio web analizado del término buscado.

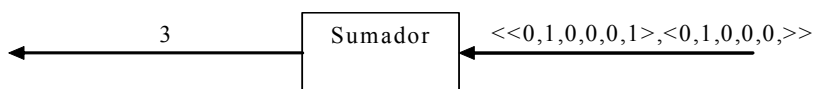


Figura 2: Visión de Caja Negra del Transformador de Streams Sumador

```

schema Sumador =

sumaAux: Int-list → Int
sumaAux (l) is
  sumaAux(<>) = 0
  sumaAux(l) = hd(l) + sumaAux(tl(l))
end,
suma: (Int-list)-list -> Int-list
suma (l) is
  suma(<>) = <>
  suma(l) = sumaAux(add(sumaAux(hd(l)),suma(tl(l))))
end
end.

```

Especificación 2: Transformador de streams Sumador

6. Conclusiones

El objetivo de nuestra investigación radica en definir una técnica que permita medir el nivel de posicionamiento de un sitio web para, de esta manera, ayudar a los desarrolladores a mejorar la calidad del mismo. En esta primera etapa de nuestra investigación, estamos definiendo un conjunto de métricas que nos permita medir el nivel de posicionamiento de un sitio dado. En el presente trabajo se presenta el resultado de aplicar los formalismos de Streams y el lenguaje de especificación del método RAISE RSL para la especificación de una de las métricas definidas. Los streams realizan nuevos aportes en el ámbito de las especificaciones, como poder analizar una componente de software teniendo en cuenta sus historias de entrada salida, y diseñar componentes de software complejas a partir de otras más simples fomentando de esta forma el reuso de componentes.

Se pudo comprobar que la traducción de las definiciones simples realizadas con streams al lenguaje RSL presentan una complejidad media; y que el verificador de tipos y los "test case" de RSL son de gran ayuda cuando se verifican y combinan las componentes.

En la continuación de nuestro trabajo se completará el conjunto de métricas y se planificará la validación teórico-práctica de las mismas, de manera de obtener un conjunto que se útil y aplicable de métricas.

Bibliografía

- [1] W. Dosch, "Scanning Stream.," presented at ACIS. International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Application, Foz do Iguazu., 2002.
- [2] R. Stephens, "A Survey of Stream Processing.," 1997.
- [3] www.iist.unu.edu, "RAISE tools."
- [4] W. Dosch, "Views of a Bounded Stack.," presented at ACIS. International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Application, Foz do Iguazu., 2002.
- [5] A. Denning, "SEO (Search Engine Optimization): A Case Study.," *The Beach Trail Cottages, Internet Marketing Newsletter*, 2003.
- [6] N. E. Fenton, *Software Metrics: A Rigorous Approach*: Chapman & Hall, Ltd., 1991.
- [7] W. Dosch, D. Riesco, M. Beron, and G. Montejano, "Formal Specification Scanning Streams using RAISE," presented at Innovations Through Information Technology, New Orleans, Louisiana., 2004.