Cahiers GUTenberg

SGMLQL : UN LANGAGE DE REQUÊTES POUR LA MANIPULATION DE DOCUMENTS SGML

Stéphane Harié, Élisabeth Murisasco, Jacques Le Maitre, Jean Véronis

Cahiers GUTenberg, nº 24 (1996), p. 181-184.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1996___24_181_0>

SgmlQL : un langage de requêtes pour la manipulation de documents SGML

Stéphane HARIÉ[a], Élisabeth MURISASCO[b], Jacques LE MAITRE[b] et Jean VÉRONIS[a]

[a] *Laboratoire Parole et Langage*
Université de Provence et CNRS
29, avenue Robert Schuman
13621 Aix en Provence Cedex 1, France
{Stephane.Harie,Jean.Veronis}
@lpl.univ-aix.fr

[b] *Groupe d'Étude du Codage de Toulon*
Université de Toulon
BP 132
83957 La Garde Cedex, France
{muri,lemaitre}
@univ-tln.fr

1. Introduction

SgmlQL est un langage de programmation développé dans le cadre du projet européen MULTEXT, permettant de manipuler des documents SGML, et en particulier des documents TEI. C'est un langage fonctionnel basé sur SQL, qui permet des manipulations complexes des documents, par exemple:

- extraction de parties de documents SGML qui satisfont des critères donnés ;
- tests, comptages et autres calculs sur les éléments SGML d'un document ;
- construction de nouveaux éléments à partir du résultat des requêtes.

2. Exemples

Quelques exemples permettront au lecteur d'avoir un aperçu des possibilités offertes par SgmlQL. Un manuel de référence complet est accessible sur le World Wide Web (voir ci-dessous).

2.1. Déclaration de variables de divers types

Exemple : affecter à la variable globale `$mybook` le document contenu dans le fichier `book.SGML` :

```
global $mybook = file "book.SGML" ;
```

2.2. Extraction d'éléments d'identificateur générique donné

Exemple : extraire le premier titre (élément `TITLE`) dans `$mybook`.

```
first TITLE of $mybook ;
```

Exemple : extraire tous les titres (du livre, des chapitres, sections, etc.) dans `$mybook`.

```
every TITLE within $mybook ;
```

2.3. Extraction d'éléments satisfaisant des critères donnés

Exemple : extraire tous les chapitres de `$mybook` qui n'ont pas de sous-section.

```
select
  first TITLE of $c
from
  $c in every CHAPTER within $mybook
where
  empty(every SECTION within $c);
```

2.4. Tests et calculs divers sur les objets

Exemple : existe-t-il un auteur dont le nom contient la chaîne `Fred` dans `$mybook`?

```
exists $a in
  (every AUTHOR within $mybook)
:
  content($a) match "Fred" ;
```

Exemple : extraire le titre de la section, le titre du chapitre et le nombre de paragraphes pour chaque section de `$mybook`.

```
select
   (first TITLE of $s) .
   (first TITLE of $c) .
    count(every PAR within $s) .
 from
    $c in every CHAPTER within $mybook,
    $s in every SECTION within $c;
```

2.5. Construction de nouveaux élements et documents SGML

Exemple : extraire toutes les entrées bibliographiques de `$mybook` et fabriquer un nouveau document de type `BIBLIO` contenant la liste de ces entrées, en plaçant le titre de `$mybook` dans un attribut `SOURCE`.

```
document
  docdtd "bib.dtd"
  with
    element BIBLIO
    attrs {SOURCE = text(first TITLE of $mybook)}
    with
       every BIBENTRY within $mybook;
```

3. Implémentation

Un interpréteur, MtSgmlQL, a été développé sous UN*X. MtSgmlQL est orienté vers la manipulation de flots, et les documents sont traités séquentiellement, comme c'est le cas dans de nombreux « filtres » UN*X et les interpréteurs de commandes tels que *awk* ou *perl*. MtSgmlQL ne nécessite aucune indexation préalable des documents et ne se préoccupe pas de leur DTD. Les documents analysés doivent simplement être du SGML valide et normalisé, c'est-à-dire sans minimisation, sans SUBDOC, sans sections marquées, etc. Les résultats sont fournis sur le flot de sortie au fur et à mesure de leur calcul, sans attendre l'analyse complète des documents si celle-ci n'est pas nécessaire. Il est donc possible d'utiliser MtSgmlQL en pipeline dans une chaîne d'outils de traitement de documents, en combinaison avec d'autres outils UN*X.

4. Information

La documentation (Référence du langage SgmlQL, et manuel de l'interpréteur) est disponible sous format navigable sur le *World Wide Web* à l'adresse :

`http://www.lpl.univ-aix.fr/projects/multext/MtSgmlQL/`

L'interpréteur peut être téléchargé gratuitement depuis la même adresse pour une utilisation non-commerciale et non-militaire (voir convention d'utilisation).

Une description du projet MULTEXT est disponible à l'adresse :

`http://www.lpl.univ-aix.fr/projects/multext/`