Raptor - использовать не будем

Краткое описание

Raptor RDF Parser Library - набор парсеров и сериалайзеров, поддерживающий RDF-триплеты и словари(из известных мне - Dublin Core, OWL). Представление - какие-то гнусные "объекты Си" - порождение извращенных фантазий какого-то фиолетового осьминога.

Утверждают, что есть отличная интеграция с Rasqal RDF Query Library(запросы к RDF) и Redland Language Bindings (API на разных языках, в т.ч. Python (смотри здесь)).

Поддержка OWL - практически отсутствует, за исключением уровня пространств имён. OWL-файлы представляются как RDF.

Пример синтаксиса

Синтаксис - своеобразный. Следующий код съест на входе файл формата RDF/XML и выведет на экран RDF-тройки.

```
#include <stdio.h>
#include <raptor.h>

/* rdfprint.c: print triples from parsing RDF/XML */

void
print_triple(void* user_data, const raptor_statement* triple)
{
  raptor_print_statement_as_ntriples(triple, stdout);
  fputc('\n', stdout);
}

int
main(int argc, char *argv[])
{
  raptor_parser* rdf_parser=NULL;
  unsigned char *uri_string;
  raptor_uri *uri, *base_uri;

  raptor_init();

  rdf_parser=raptor_new_parser("rdfxml");

  raptor_set_statement_handler(rdf_parser, NULL, print_triple);

  uri_string=raptor_uri_filename_to_uri_string(argv[1]);
  uri=raptor_new_uri(uri_string);
  base_uri=raptor_uri_copy(uri);
```

```
  raptor_parse_file(rdf_parser, uri, base_uri);

  raptor_free_parser(rdf_parser);

  raptor_free_uri(base_uri);
  raptor_free_uri(uri);
  raptor_free_memory(uri_string);

  raptor_finish();
}
```

Вердикт

Если хватает RDF-a, то Redland, Rasqal и Raptor будут хорошим выбором.

Если нужно представление в виде OWL, то не подходит.

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/etc:teach:diplomants:projects:2009:olpcmind:links:semantic_tools:frameworks:raptor

Last update: **2008/07/09 02:43**