

Intuicyjny język wyszukiwania TQL Tablets Query Language

Joanna Kotuła, Aleksandra Murawska

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

22 października 2009

Spis treści

- 1 Domain Specific Languages
- 2 Tabliczki sumeryjskie
- 3 Tablets Query Language
- 4 Struktura translatora
- 5 Co zrobiliśmy, co planujemy

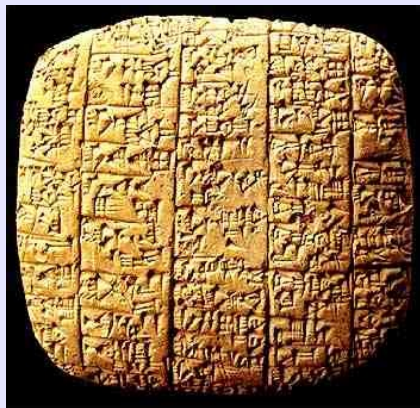
Domain Specific Languages (DSL)

DSL to języki przystosowane do używania w konkretnym celu.

- łatwość użycia
- specyficzne konstrukcje
- ograniczone możliwości
- kompilacja do języka niższego poziomu (np. SQL)

Tabliczki sumeryjskie

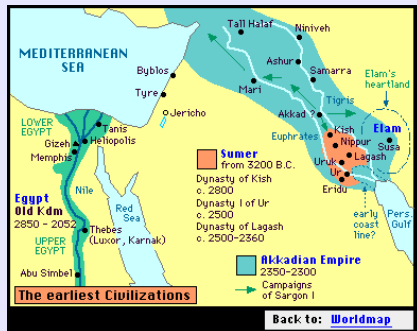
- gliniane tabliczki pokryte pismem klinowym
- pochodzą z terenów Bliskiego Wschodu
- najstarsze znane pismo (powstało ok. 3500r. p.n.e.)
- pismo początkowo rysunkowe, później coraz prostsze
- głównie teksty gospodarcze i administracyjne



Sumerologia

Sumerologia to nauka badająca kulturę i historię starożytnych Sumerów, czerpiąca wiedzę m.in. z zachowanych tabliczek.

- zdigitalizowane i ręcznie skorygowane treści tabliczek są udostępnione przez system CDLI (obecnie prawie 225 000 tabliczek, nam się udało ściągnąć 147 468)
- możliwość niewłaściwej interpretacji klinów
- potrzebne intuicyjne narzędzie do wyszukiwania w bazie tabliczek na podstawie odczytów



Tablets Query Language

- Język pozwalający sumerologom łatwo tworzyć zapytania w bazie tabliczek.
- Prosta składnia zapytań.
- Zapytania zawierają informacje tylko o treści tabliczek i o ich metadanych.
- Możliwość stworzenia implementacji na dowolną bazę przechowującą tabliczki.

Przykładowe zapytania TQL

przykład 1

provenience: Ur
period: "Ur III"
genre: Administrative
text: udu + (masz2/ugula) —szabra

Przykładowe zapytania TQL

przykład 2

```
define
  provenience: Ur
  period: "Ur III"
  genre: Administrative
  text: udu ——szabra
  as "moje zapytanie"
```

```
search
  text: masz2/ugula
  in "moje zapytanie"
```


Przykładowe zapytania TQL

przykład 3

provenience: Ur
period: "Ur III"
genre: Administrative
text: udu ——szabra

text: masz2/ugula
provenience: Ur

Przykładowe zapytania TQL

przykład 4

provenience: Ur*
period: "Ur III"
genre: Administrative
text: "udu niga" /nig2*sze3

Parser i analizator kontekstowy

- moduły niezależne od bazy danych i języka wyszukiwania
- wspólne dla wszystkich implementacji TQL
- wejście - zapytanie w języku TQL
- wyjście - drzewo składni abstrakcyjnej

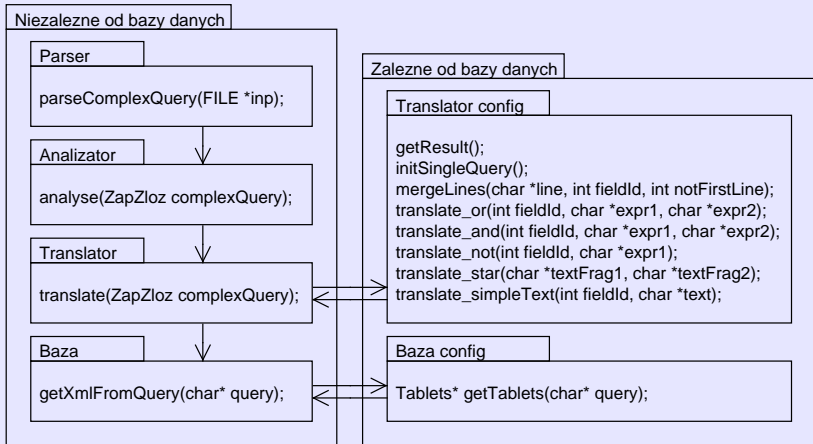
Translator

- składa się z dwóch modułów:
 - funkcje tłumaczące konkretne elementy składni
 - szkielet niezależny od języka docelowego
- wejście - drzewo składni abstrakcyjnej
- wyjście - zapytanie w odpowiednim języku

Baza

- również składa się z dwóch modułów:
 - wykonujący zapytanie i przekazujący wynik w określonej postaci
 - tłumaczący wynik na XML
- wejście - zapytanie w języku docelowym
- wyjście - XML zawierający informacje o znalezionych tabliczkach i ich treść

Podział na moduły



Aplikacja wykorzystująca translator

- umożliwia użytkownikowi wprowadzenie zapytania jako tekstu lub za pomocą graficznego interfejsu
- odpowiednio wyświetla zwracanego XML-a, pokazując wyszukiwane sekwencje

Do tej pory

- przykładowa baza relacyjna (Postgres)
- gotowy parser i analizator składniowy
- moduł translatora i moduł konfiguracyjny dla bazy relacyjnej
- moduł bazy: połączenie z bazą, wykonywanie zapytań i zwracanie XML-a – jeszcze bez zaznaczonych sekwencji wyszukiwania
- wszystko napisane w C (+ BNF-converter)

Plany

- zaznaczanie w XML-u z tabliczkami wyszukiwanych sekwencji
- interfejs graficzny www – do wprowadzania zapytań i wyświetlania wyników
- moduły konfiguracyjne do innej bazy danych
- optymalizacje zapytań (obowiązkowo dla negacji)
- tłumaczenie odczyty \leftrightarrow kliny i wyszukiwanie po klinach

Co jeszcze można zrobić

- inteligentny interfejs tekstowy
 - podpowiadanie przy tekstowym wpisywaniu do edytora zapytań
 - skróty (prov zamiast provenience)
- forma wyniku do wyboru
- aplikacja desktopowa
- wyszukiwanie po tagach, kodach tematycznych, peselach

Dziękujemy za wysłuchanie.

Pytania? Uwagi?