Intuicyjny język wyszukiwania TQL Tablets Query Language

Joanna Kotuła, Aleksandra Murawska

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

19 października 2009

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)

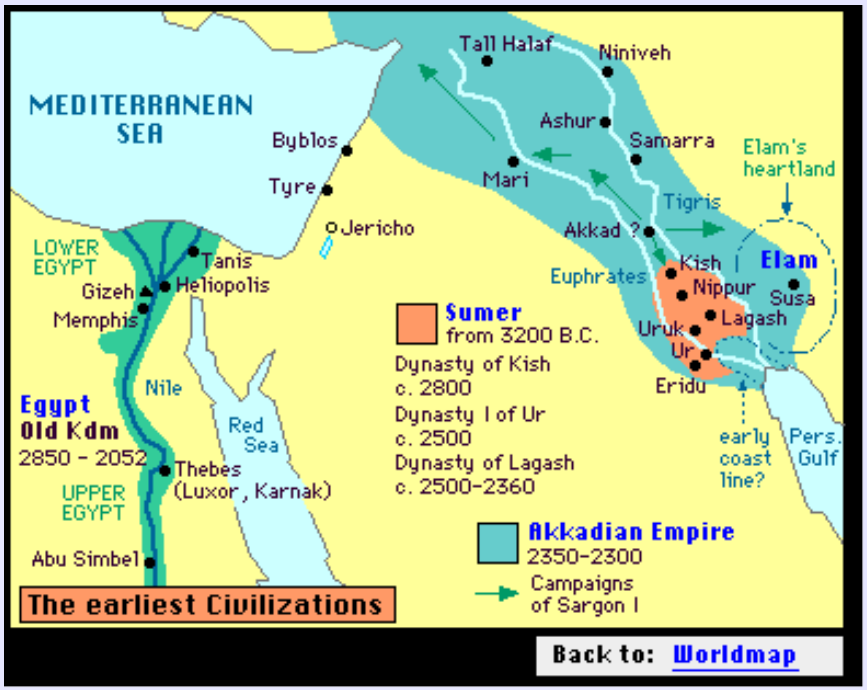
Tabliczki sumeryjskie

- 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)
- 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 ich metadanych.
- Możliwość stworzenia implementacji na dowolną bazę przechowującą tabliczki.

Przykładowe zapytania TQL

przykład 1

provenience: Ur*
period: "Uruk III"
genre: Administrative
text: udu + (szid / sipa) −− adad-tilati

Przykładowe zapytania TQL

przykład 2

provenience: Gar*
period: UrIII
genre: Administrative
text: udu/masz2

provenience: Ur
period: UrIII
text: sig4

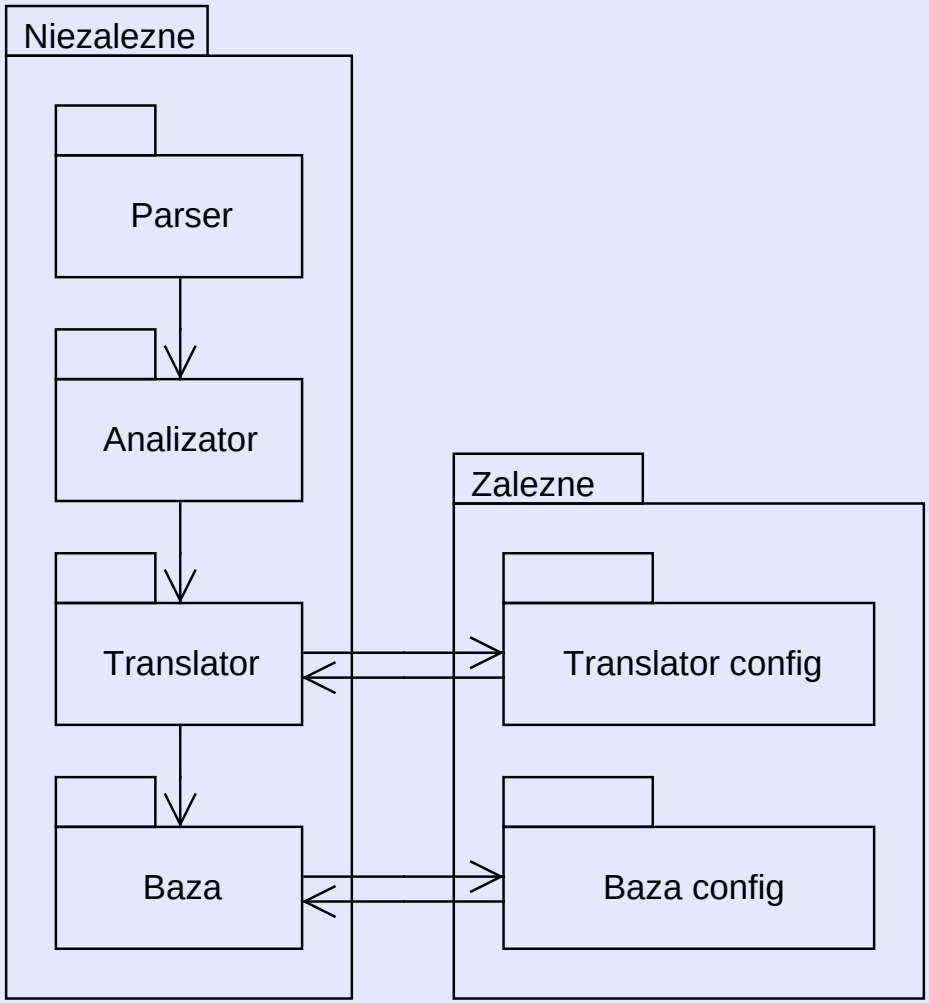
Przykładowe zapytania TQL

przykład 3

define
provenience: Garshana
period: UrIII
text: "udu ban"/mash2
as "zwierzaki w Garshana"

search
text: adad-tilati
in "zwierzaki w Garshana"

Podział na moduły



Parser i analizator składniowy

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

Translator

- moduł zależny od języka docelowego
- komunikuje się z modułem konfiguracyjnym, który zawiera funkcje tłumaczące poszczególne konstrukcje
  - zapytanie
  - linia zapytania
  - i $(+)$
  - lub $(/)$
  - nie $(--)$
  - cokolwiek $(*)$
- wejście - drzewo składni abstrakcyjnej
- wyjście - zapytanie w odpowiednim języku

Baza

- moduł zależny od wykorzystywanej bazy danych
- komunikuje się z modułem konfiguracyjnym, który zawiera funkcję wykonującą zapytanie
  - moduł konfiguracyjny zawiera oddzielny plik z danymi dostępu do bazy
- przetwarza wynik zapytania do postaci XML
- wejście - zapytanie w języku docelowym
- wyjście - XML zawierający informacje o znalezionych tabliczkach i ich treść

Aplikacja wykorzystująca translator

- umożliwa 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

Plany

- zaznaczanie w XML-u z tabliczkami wyszukiwanych sekwencji
- interfejs graficzny – 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