DAR

Výzkumné centrum Data – Algoritmy – Rozhodování

Research Centre Data – Algorithms – Decision Making

Mgr. Jan Peroutka
Pavel Kotyza
Ing. Daniela Antoňová
JUDr. Ivan Gabaš

Linguistic database - SDK description

Interní publikace DAR – DELTAX SYSTEMS A.S. 2005/3
Praha, prosinec 2005

Linguistic database - SDK description

- Set of documents describing the SDK and FullText properties and usage:
    - This documentation
- Demo programs for:
    - Morphology routines:
        - **MORPHOLO.cpp** - source text - performs single functions on a number of typical examples
        - **MORPHOLO.dsp** - project
        - **MORPHOLO.dsw** - project
- Program files (they do not require any maintenance by application programmers):
    - **Lv_dll.dll** - library performing single functions
    - **Lv_dll.lib** - library serving to correct linking of application program
    - ***.h** - modules serving to correct compilation of application program (the modules are targeted to VisualBasic and Visual C 6.0++; for Java and C# the type **long** should be modified to **int**)
- Data files (they do not require any maintenance by application programmers):
    - **Liv_data.do1** - base ZD - data
    - **Liv_data.dd1** - base ZD - index
    - **Liv_data.ro2** - additions ZD - data
    - **Liv_data.rd2** - additions ZD - index
    - **Liv_data.dsd** - similar words (e.g. for CS-EN dictionary) - optional
    - **Liv_data.dms** - similar words taken from multilingual word corpus - optional
    - **Liv_data.len** - literals
- Dictionary files - they are optional, depending on user license. For each dictionary there are five of them:
    - name1.xml - data file
    - name2.xxi, name2.xxd, name2.dsd, name2.tlo - index files

General description

- SDK is written in 'C' and compiled under Win32 (MSVC 6.0). The function calls are very simple. Nevertheless it is necessary to care for different environments, especially that for C# the type **long** is 64 bits.
- Character coding:
    - Most of the programming interface works with UTF-8 coding.
    - In fulltext routines the coding is detected automatically or can be set explicitly.
- Parameters of languages - they are given as 2 bytes abbreviations from ISO 639, i.e. "CS", "EN", "DE", "JA", "RU", "SK", "ZH" and so on.
- Word separators are space, comma, semicolon, colon, dot, exclamation mark, question mark.
- The program is able to process universal affixes. Currently it works for:
    - language CS: **ne-**, **nej-**
- Single routines return wide range of values. It is up to application programmer to filter out the values not useful for end user.
- The whole system is extensible. In the future its services will include:

    - complex grammar info about any word,
    - translations to other languages.
- The Morphological database is large, nevertheless, due to immense dimensions of natural languages, it is not complete. It must be taken into account when programming its applications.
- Fulltext indexing and retrieval functions are just basic functions for simple processing of single file. The SDK can be extended by more functions allowing for indexing of whole directories and for more sophisticated data retrieval.
- All functions return an error code:
    - 0 for normal return,
    - nonzero code indicates an error found, the error mechanism is either regular return thru program stack or stack bypassed by longjmp() routine.

Overview of functions

Initialization

- long DB_SimpleInitialize(char *path)
    - Initialization routine for the Database. It should be called once, at the program start. If using fulltext routines, please, use FULLTEXT_Start().
    - path - parameter contains the path to Database files, e.g. **D:\Morphologica**.
    - Return values:
        - 0 - O.K.
        - other - error

- long DB_SimpleShutDown(void)
    - Closing routine for Database. It should be call once, at program end. If using fulltext routines, please, use FULLTEXT_Stop().
      Improper program end does not harm the system.
    - Return value:
        - 0 - O.K.
        - other - error

Querying morphology

- long MORPHO_GetAllShapes(char *word_in,
                                              char *language_in,
                                              char *shapes_out,
                                              long shapes_out_max_len,
                                              short variate_case)
    - The routine returns all shapes derivable from word_in. The shapes are sorted in ascending order. The routine is suitable for fulltext search for word.
    - word_in - input word
    - language_in - see description
    - shapes_out - shapes separated by spaces
    - shapes_out_max_len - length for shapes_out - recommended length is 16 KB

    - variate_case - 0 => no, 1 => yes, i.e.. shapes_out will contain also shapes converted to upper case and shapes converted to first letter uppecase
    - Return values:
        - 0 - O.K.
        - -1 - unexpected language abbreviation
        - -2 - shapes_out_max_len is too small

- long MORPHO_GetBasicShapes(char *word_in,
  char *language_in,
  char *shapes_out,
  long shapes_out_max_len)
    - The routine returns basic shapes of word_in. The routine is suitable for search for entries in a language dictionary. The number of basic shapes returned may vary:
        - 1 - this shape should be searched for in dictionary
        - >1 - the word_in may be homonymous with a number of basic shapes, e.g. word **ceiling** is derived from both **ceiling** and **ceil**. In this case all basic shapes should be searched for in dictionary
        - 0 - the word_in may be mistyped of not present in Database. The function MORPHO_GetSimilarWords() should be used then
    - word_in - input word
    - language_in - see description
    - shapes_out - basic shape, there may be a number of them separated by spaces. The shapes should be tested againsts the list of dictionary entries. The matches should be offered to user.
    - shapes_out_max_len - length for shapes_out - recommended length is 4 KB
    - Return values:
        - 0 - O.K.
        - -1 - unexpected language abbreviation
        - -2 - word_in not found
        - -3 - internal error
        - -4 - memory allocation error
        - -5 - shapes_out_max_len is too small

- long MORPHO_GetMorphologyList(char *word_in,
  char *language_in,
  char *shapes_out,
  long shapes_out_max_len)
    - The routine returns all grammar senses of word_in. The routine is suitable for syntactic text analysis.
    - word_in - input word, e.g. **silnicemi**
    - language_in - see description - it can be NULL or "" - then all languages are eligible.
    - shapes_out - CRLF (i.e. "\r\n") separated shape descriptions. Each description contains:
        - unique database code as 8B, e.g. **00051362**

        - full name of language, e.g. **CZECH** - the names are arbitrarily chosen for this machine processing of language - they need not match common grammar conventions
        - word stem, e.g. **silnic**
        - pattern name, e.g. **CZ*lednice**
        - logical description, e.g. **SUBST+FEM+PL+7P**
        - first (also called basic) shape in morphology of word_in, e.g. **silnice**
        - found word shape (mostly it is identical with word_in), e.g. **silnicemi**
    - shapes_out_max_len - length for shapes_out - recommended length is 64 KB
    - Return values:
        - 0 - O.K.
        - -1 - unexpected language abbreviation
        - -2 - word_in not found
        - -3 - shapes_out_max_len is too small

- long MORPHO_GetDiacriticisedWords(char   *word_in,
  char   *language_in,
  char   *words_out,
  long   words_out_max_len)
    - The routine returns all shapes diacritically related to word_in. The routine is quite slow as it searches whole Database - for longer words it can take seconds. It can return a number of shapes, some of them may be rare or obsolete, e.g. **ryze** => **ryze ryzé rýze rýže**. In this case it is necessary to offer the shapes to user so he could choose from them.
      If faster routine not returning uncommon words is required, then it is possible to use **MORPHO_GetSimilarWords()** with parameter take_diacritical_variants_only=1.
    - word_in - input word
    - language_in - see description
    - words_out - space separated shapes
    - words_out_max_len - length for words_out - recommended length is 64 KB
    - Return values:
        - 0 - O.K.
        - -1 - unexpected language abbreviation
        - -2 - words_out_max_len is too small

How to use the SDK

- The SDK functions are supplied as minimal interface, i.e. it provides access to all data with minimal set of functions and parameters. This also means that any sorting, selection and/or formatting of data returned is up to application programmer.

Výzkumné centrum Data – Algoritmy – Rozhodování

Výzkumné centrum Data – Algoritmy – Rozhodování (DAR) bylo založeno v roce 2005 v rámci programu MŠMT Výzkumná centra PP2 – DP01 (č. p. 1M6798555601; CEP 1M0572) těmito subjekty:

- Ústav teorie informace automatizace Akademie věd ČR
- Ostravská univerzita v Ostravě, Ústav pro výzkum a aplikace fuzzy modelování
- Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství
- Západočeská univerzita v Plzni, Fakulta aplikovaných věd
- Empo Praha, spol. s r. o.
- Compureg Plzeň, s. r. o.
- ELTODO, dopravní systémy, s. r. o.
- OASA COMPUTERS, s. r. o.
- DELTAX Systems, a. s.

Ediční řada **Interní publikace DAR** je určena pro rychlé předávání poznatků vznikajících v rámci činnosti Výzkumného centra Data – Algoritmy – Rozhodování. Obsahuje rukopisy článků a příspěvků na konference, výzkumné zprávy, dokumentaci pořádaných odborných akcí a další pracovní materiály s omezenou distribucí. Autoři plně odpovídají za obsah jejich textů.

Research Centre Data – Algorithms – Decision Making

Research Centre Data – Algorithms – Decision Making was established in 2005 due to support program of Ministry of Education, Youth and Sports by following institutions:

- Institute of Information Theory and Automation of the Academy of Sciences of the Czech Republic
- University of Ostrava, Institute for Research and Applications of Fuzzy Modelling
- Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering
- University of West Bohemia, Faculty of Applied Science
- Empo Praha, spol. s r. o.
- Compureg Plzeň, s. r. o.
- ELTODO, dopravní systémy, s. r. o.
- OASA COMPUTERS, s. r. o.
- DELTAX Systems, a. s.

Reports series **Interní publikace DAR** is intended for a quick transfer of knowledge produced by activites of Research Centre Data - Algorithms - Decision Making. It includes manustripts of papers and conference contributions, research reports, documentations of organised scientific events and other working prints with limited distribution. The autors are fully responsible for contents of their texts.
