

Podręcznik użytkownika programu LMFEditor

(wersja 1.8)

Przeznaczenie programu LMFEditor

Program przeznaczony jest do edycji tworzonego w ramach projektu SYNAT elektronicznego słownika dawnej polszczyzny. Słownik ten zapisany jest w pliku XML i reprezentowany przez strukturę LMF (Lexical Markup Framework). Program ten nie jest uniwersalnym narzędziem przeznaczonym do edycji dowolnych struktur LMF, lecz tylko tego ich podzbioru, który został wybrany do reprezentowania tworzonego słownika.

Danymi wejściowymi programu są dwa pliki: plik XML zawierający strukturę LMF słownika oraz plik tekstowy, na podstawie którego ta struktura powstała. Ten drugi plik jest opcjonalny, ale niewątpliwie ułatwia proces weryfikacji i edycji słownika. Program modyfikuje tylko dane z pliku XML pozostawiając dane z pliku tekstowego w niezmienionej postaci.

Graficzny interfejs użytkownika

Program posiada bardzo prosty graficzny interfejs. Posiada on 7 komponentów przedstawionych na rys. 1:

1. Główne menu
2. Pasek narzędzi
3. Indeks haseł słownika
4. Lista identyfikatorów haseł powiązanych z aktualnie wybranym hasłem
5. Tekst źródłowy na podstawie którego powstała struktura LMF hasła
6. Struktura LMF aktualnie wybranego hasła
7. Pasek informacyjny

Wszystkie narzędzia służące edycji słownika znajdują się w głównym menu programu (1). Dla wygody, niektóre z nich zostały umieszczone na pasku narzędzi (2).

Indeks (3) jest listą haseł słownika. Hasła w indeksie podzielone są na cztery grupy. Pierwszą grupę stanowią hasła zaznaczone na liście czcionką w czarnym kolorze. Są to hasła proste (najczęściej jednowyrazowe), których opis w pliku tekstowym pasował do konwencji stosowanej w słowniku w wersji papierowej. Niestety, konwencje te łamane były wielokrotnie. Poza tym wiele błędów w wersji tekstowej zostało wprowadzonych przez program OCR. Stąd pojawiła się grupa haseł (*unprocessed* – czcionka czerwona), których automatyczne przetworzenie nie było możliwe oraz takie (*problematic* – czcionka niebieska), co do których istnieje obawa, że zostały błędnie przetworzone do postaci LMF. Istnieje ponadto jeszcze jedna grupa haseł (*automatically added* – czcionka zielona) zawierająca hasła będące częściami składowymi haseł złożonych, a które nie występują w słowniku, albo występują w innej, odmienionej formie.

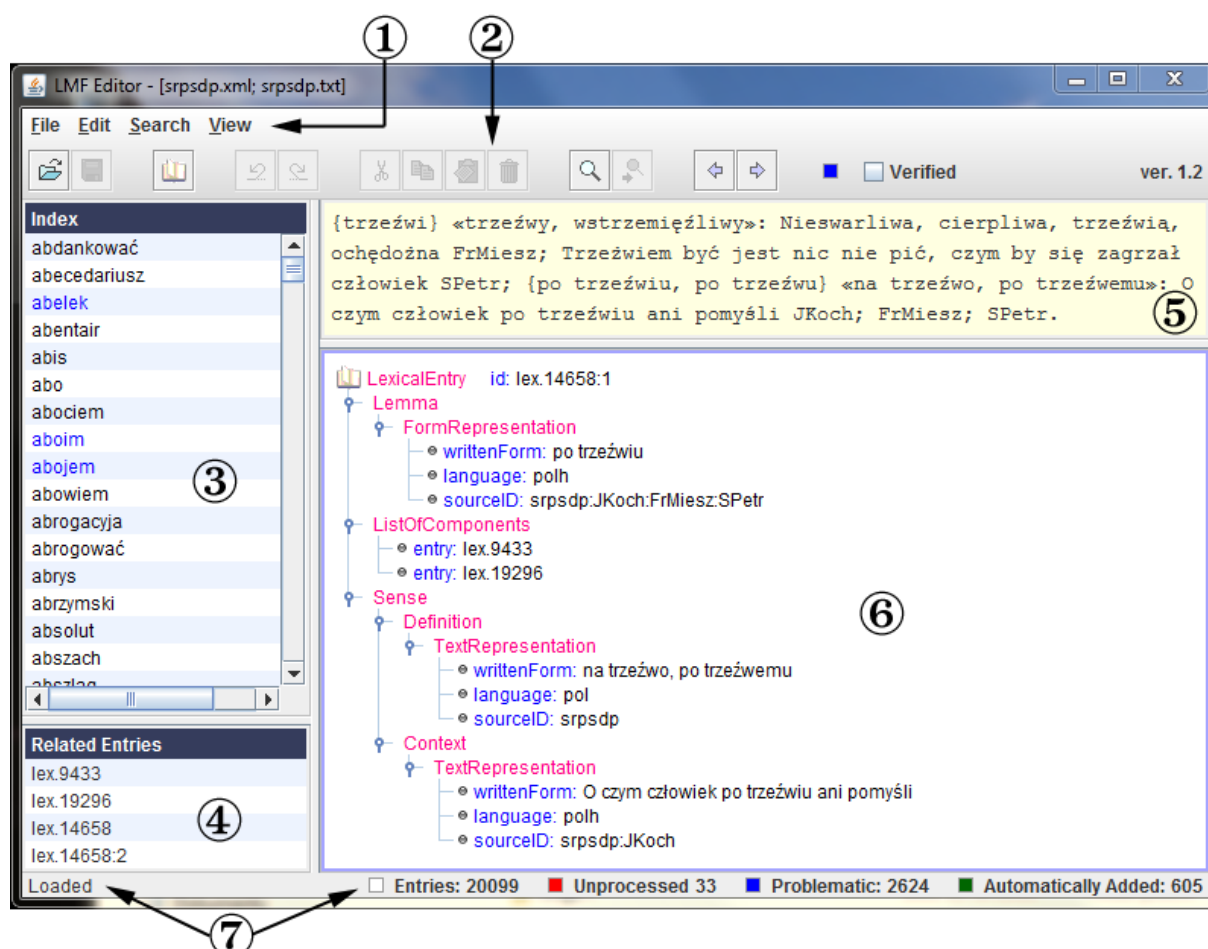
Hasła z indeksu wybieramy przy pomocy myszki lub klawiatury (klawisze ↑↓). Wybranie hasła spowoduje wyświetlenie struktury LMF hasła (6), tekstu źródłowego (5), na podstawie którego ta struktura powstała oraz listy identyfikatorów (4) struktur związanych z aktualnie wybranym hasłem.

Program umożliwia usuwanie haseł z indeksu przy pomocy polecenia **Delete** z menu **Edit**.

W polu wyświetlającym tekst źródłowy (5) użytkownik może zaznaczyć interesujący go fragment tekstu, skopiować go i następnie wkleić w odpowiednie miejsce w strukturze LMF.

Struktura LMF (6) jest strukturą drzewiastą. Nazwy poszczególnych węzłów drzewa są wyświetlane w kolorze ciemnoróżowym. Nazwy liści nie są wyświetlane. Liśćmi drzewa mogą być jedynie dwie podstruktury LMF: <feat> oraz <Component>. W przypadku tylko tych podstruktur, w celu uzyskania większej przejrzystości, wyświetlane są tylko ich atrybuty. Nazwy atrybutów zaznaczone są czcionką niebieską, zaś ich wartości, czcionką czarną.

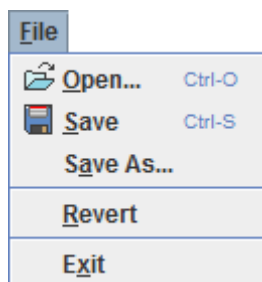
Pasek informacyjny (7) zawiera różnego rodzaju komunikaty oraz statystykę dotyczącą liczby haseł w różnych ich grupach.



Rys. 1. Główne okno edytora

Strukturę LMF (6) można edytować albo przy pomocy menu **Edit**, albo przy pomocy menu kontekstowego wyświetlanego po naciśnięciu prawego klawisza myszy w momencie, gdy kursor wskazuje na któryś z węzłów drzewa. W tym pierwszym przypadku należy najpierw uaktywnić polecenia z menu **Edit** poprzez wybranie struktury, którą chcemy edytować. Strukturę można wybrać poprzez kliknięcie lewym klawiszem myszy lub przy pomocy klawiatury (klawisze ↑↓). Możemy wtedy usunąć wskazywaną strukturę, dodać do niej inną strukturę, dodać lub usunąć atrybut. Zarówno nazwa jak i wartość atrybutu też mogą być edytowane. W tym celu należy dwukrotnie kliknąć na nazwę lub wartość atrybutu. W przypadku nazwy atrybutu pojawi się wówczas lista zawierająca wszystkie dozwolone nazwy, z której należy wybrać jedną. W przypadku, gdy na liście nie ma nazwy, którą użytkownik chciałby wybrać, program umożliwi utworzenie nowej. W przypadku wartości atrybutów mogą zachodzić dwa przypadki: wartość jest dowolnym tekstem, lub elementem określonego zbioru wartości. W pierwszym przypadku, pojawi się edytowalne pole tekstowe. W drugim przypadku, podobnie jak dla nazw atrybutów, pojawi się lista możliwych wartości do wyboru, którą także można uzupełnić o wartość, która na niej nie występuje. Przy edytowaniu nazw i wartości atrybutów dostępne są standardowe polecenia edytujące: **Cut**, **Copy**, **Paste**, **Delete**, **Undo** i **Redo**.

Menu programu



Po pierwszym uruchomieniu programu należy wczytać do pamięci strukturę słownika. Służy temu polecenie **Open**. Po wybraniu tego polecenia z menu wyświetli się standardowe okno dialogowe, w którym należy wybrać odpowiedni plik o rozszerzeniu .xml. Następnie, po raz kolejny program poleci wybrać z listy plików jeden element (zazwyczaj o tej samej nazwie, ale z rozszerzeniem .txt) zawierający tekst, na podstawie którego utworzono strukturę LMF słownika. Wybór tego pliku jest opcjonalny, ale ułatwia on poprawianie struktury LMF zawartej w pliku .xml.

Przy następnym wywołaniu programu, oba wybrane pliki są wczytywane automatycznie.

Polecenia **Save** i **Save As...** służą zapisaniu na dysku wprowadzonych poprawek.

Polecenie **Revert** służy otworzeniu danych do postaci w jakiej one były po ostatnim wykonaniu polecenia **Save**.

Pracę programu można zakończyć zamykając jego główne okno lub wybierając z menu polecenie **Exit**. Zanim program zakończy pracę, poprosi użytkownika aby zapisał wprowadzone poprawki, o ile takie poprawki zostały dokonane. Następnie zapamięta na dysku różne ustawienia programu: ścieżki

plików wejściowych, parametry wyszukiwania oraz parametry sortowania i wyświetlania indeksu słownika. Informacje te zapisane są w prywatnym katalogu użytkownika w pliku LMFEEditorPrefs.

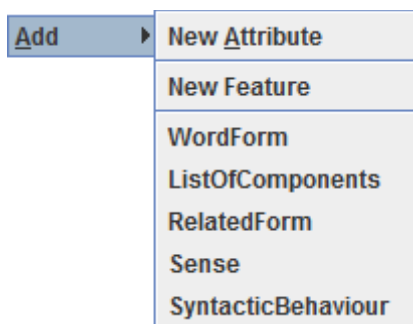


Polecenie **Undo** (cofnij modyfikację) i **Redo** (zastosuj ponownie) odpowiednio przywraca stan sprzed ostatniej modyfikacji i dokonuje ponownej modyfikacji po tym jak użytkownik wykonał polecenie **Undo**. Program pamięta 20 ostatnio wykonywanych modyfikacji.

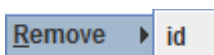
Standardowe polecenia **Cut** (wytnij), **Copy** (kopiuj), **Paste** (wklej) oraz **Delete** (usuń) są aktywne lub nieaktywne w zależności od tego, który z elementów interfejsu jest w danej chwili aktywny.

Poleceniem **Verified** zmieniamy status danego hasła. Jeśli hasło należało do jednej z grup: *unprocessed*, *problematic* lub *automatically added*, to wykonaniu tego polecenia staje się zweryfikowanym. W przeciwnym przypadku hasło jest zakwalifikowane do grupy hasel *problematic*.

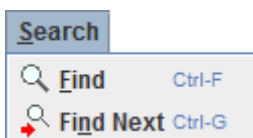
Nowe hasło główne wstawiamy do słownika przy pomocy polecenia **New Lexical Entry**. Nowa struktura otrzyma automatycznie unikalny identyfikator.



Menu **Add** jest tworzone dynamicznie i zależy o tego jaka podstruktura LMF jest w danej chwili wybrana. Powyższy przykład dotyczy przypadku, w którym wybrana została struktura odpowiadająca całemu hasłu. Można więc dodać nowy atrybut (**New Attribute**) dla tej struktury oraz nową własność (**New Feature**). Można także dodać podstruktury (<WordForm>, <ListOfComponents>, <RelatedForm>, <Sense> i <SyntacticBehaviour>). W menu umieszczone są tylko nazwy tych podstruktur, które mogą być dodane do aktualnie wybranej struktury.

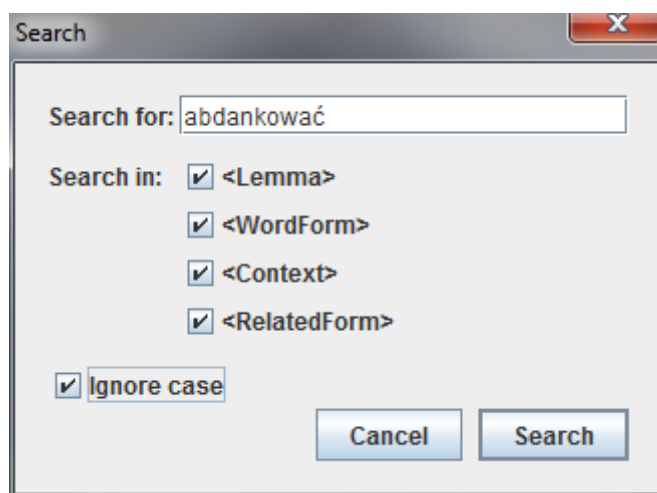


Podobnie jak w przypadku menu **Add**, menu **Remove** jest tworzone dynamicznie. Na liście tego menu znajdują się atrybuty aktualnie wybranej struktury, które możemy usunąć.

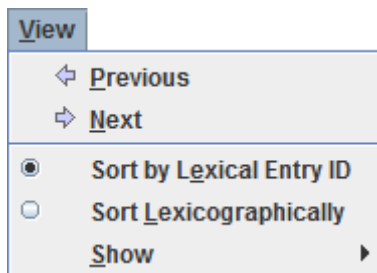


Po wybraniu z menu polecenia **Find** program wyświetla dialog (rys. 2). Dialog ten pozwala użytkownikowi na ustalenie pewnych parametrów wyszukiwania tekstu w słowniku. Szukany tekst jest reprezentowany przez wyrażenie regularne, które należy wpisać w polu **Search for** dialogu. Użytkownik może wyspecyfikować do których fragmentów opisów haseł słownika należy ograniczyć wyszukiwanie. Może być to forma podstawowa (<Lemma>), zbiór form wyrazowych (<WordForm>), zbiór cytatów z tekstów źródłowych (<Context>) lub zbiór form pokrewnych (<RelatedForm>). Wyszukiwanie może brać pod uwagę lub ignorować wielkość liter w tekście. Wyszukiwanie zawsze rozpoczyna się od pierwszego hasła w słowniku niezależnie od wybranej metody sortowania. Program wyświetla pierwszą strukturę zawierającą szukane wyrażenie lub informuje użytkownika o niepowodzeniu przy pomocy sygnału dźwiękowego. W wyświetlanych strukturach miejsca odpowiadające szukanemu wyrażeniu są zaznaczone kontrastowym żółtym kolorem.

Polecenie **Find Next** staje się dostępne w przypadku powodzenia polecenia **Find**. Słownik przeszukiwany jest wtedy od ostatnio znalezionej struktury aż do końca słownika a potem z powrotem od jego początku.

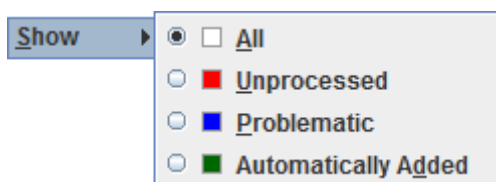


Rys. 2 Dialog wyszukiwania tekstu



Polecenia **Previous** i **Next** służą do nawigowania po ostatnio wyświetlanych strukturach. Działają one podobnie jak w przypadku przeglądarek internetowych, które umożliwiają szybki powrót do poprzednio odwiedzanych stron. Program pamięta 20 ostatnio wyświetlanych struktur.

Hasła słownika mogą być sortowane na dwa sposoby: po identyfikatorze hasła (**Sort by Lexical Entry ID**) lub alfabetycznie (**Sort Lexicographically**). W przypadku pierwszej metody sortowania hasło główne i jego podhasła będą się znajdować obok siebie na liście indeksu.



Wyświetlany indeks haseł można zawęzić tylko do jednej ze wspomnianych na wstępie grup. Domyślnie program wyświetla pełen indeks.

Wymagania sprzętowe

Program napisany jest w języku Java i pracuje pod dowolnym systemem, na którym zainstalowana jest Java w wersji 1.7 lub wyższej. Program potrzebuje do pracy ok. 500MB pamięci.