

**Uniwersytet Warszawski**  
Wydział Matematyki, Informatyki i Mechaniki

**Krzysztof Dudzik**

Nr albumu: 248349

**Aplikacja wspomagająca  
tworzenie i edycję haseł  
w polskim Wikisłowniku**

**Praca magisterska  
na kierunku INFORMATYKA**

Praca wykonana pod kierunkiem  
**dr. hab. Jerzego Tyszkiewicza, prof. UW**  
Instytut Informatyki

2011

## **Oświadczenie kierującego pracą**

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

## **Oświadczenie autora pracy**

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora pracy

## **Streszczenie**

Tematem pracy jest aplikacja służąca do ułatwienia pracy autorów haseł w polskim Wikisłowniku. Jej funkcje mają w maksymalny możliwy sposób ułatwić tworzenie i edytowanie haseł osobom bez wiedzy informatycznej i technicznej, a także automatyzować możliwie wiele rutynowych czynności wykonywanych przy redagowaniu hasła, jak tworzenie łączy do haseł powiązanych, zautomatyzowane szukanie zapisów i nagrań wymowy, grafik, przykładów użycia, wystąpień w związkach frazeologicznych itp. Dodatkowo aplikacja może przejąć część funkcji realizowanych obecnie za pomocą botów.

## **Słowa kluczowe**

Wikisłownik, Fundacja Wikimedia, MediaWiki, wiki, edytor, API, JavaScript, jQuery, interfejs użytkownika, społeczność internetowa, open source

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

11.3 Informatyka

## **Klasyfikacja tematyczna**

H. Information Systems

H.5. Information Interfaces and Presentation

H.5.3. Group and Organization Interfaces

## **Tytuł pracy w języku angielskim**

An application supporting article creation and edition for the Polish Wiktionary



# Spis treści

<b>1. Wprowadzenie</b>	<b>5</b>
<b>2. Wikisłownik</b>	<b>7</b>
2.1. Projekty Fundacji Wikimedia	7
2.2. Oprogramowanie MediaWiki	9
2.2.1. Edytowanie i wikitekst	10
2.3. Wiktory – Wikisłownik	12
2.4. Polska edycja Wikisłownika	14
2.4.1. Struktura hasła	15
2.4.2. Wady dotychczasowych rozwiązań	26
2.4.3. Różnice w stosunku do Wikipedii	29
<b>3. Aspekty społecznościowe</b>	<b>31</b>
3.1. Koncepcja <i>wiki</i>	31
3.2. Społeczność polskiej edycji Wikisłownika	32
3.3. Analiza wymagań	34
3.3.1. Definicje	34
3.3.2. Wymagania funkcjonalne	35
3.3.3. Wymagania нефункционалне	36
3.4. Specyfika tworzenia aplikacji dla wikispołeczności	38
<b>4. Opis implementacji</b>	<b>39</b>
4.1. Wprowadzenie	39
4.1.1. JavaScript w Wikisłowniku	40
4.1.2. Środowisko programistyczne	41
4.1.3. Dobre praktyki	42
4.2. Formularz edycyjny	44

4.2.1. Interfejs użytkownika . . . . .	45
4.2.2. Parsowanie i drukowanie wikitekstu . . . . .	51
4.3. Automatyzacja edycji hasła . . . . .	54
4.3.1. API MediaWiki . . . . .	54
4.3.2. Funkcje automatyzujące edycję haseł . . . . .	57
4.4. Wdrożenie i dalszy rozwój . . . . .	65
<b>5. Podsumowanie . . . . .</b>	<b>67</b>
<b>A. Zawartość płyty CD . . . . .</b>	<b>69</b>
<b>B. Wygląd aplikacji — zrzuty ekranu . . . . .</b>	<b>71</b>
<b>Bibliografia . . . . .</b>	<b>73</b>
<b>Spis ilustracji . . . . .</b>	<b>77</b>
<b>Spis tabel . . . . .</b>	<b>77</b>
<b>Spis listingów . . . . .</b>	<b>77</b>

# Rozdział 1

## Wprowadzenie

Żyjemy w czasach, w których nieustannie zmienia się sposób wyszukiwania informacji przez przeciętnego człowieka. Z roku na rok coraz mniejszą rolę odgrywają papierowe kompendia takie jak encyklopedie i słowniki, stopniowo przybierają natomiast na znaczeniu elektroniczne bazy wiedzy — szczególnie zaś internetowe zbiory danych. Przyczyny tego stanu rzeczy są oczywiste: chodzi przede wszystkim o wygodę korzystania ze stron internetowych. Brak możliwości wyszukiwania w obrębie ogromnych ilości danych powoduje, że encyklopedie i słowniki w postaci książek stają się o wiele mniej atrakcyjne dla kogoś, kto chce zdobyć nowe informacje.

Wszechobecny dostęp do internetu sprawia, że to właśnie w sieci WWW powstają najbardziej popularne bazy ludzkiej wiedzy. Nie ma chyba internautów, którzy nie korzystaliby, rzadziej lub częściej, z Wikipedii — internetowej encyklopedii pisanej przez ochotników. Właśnie fakt, że encyklopedia ta współtworzona jest przez amatorów, stanowi o jej wyjątkowym charakterze, który zostanie w tej pracy pokrótce opisany. Wikipedia stale utrzymuje się w pierwszej dziesiątce najczęściej odwiedzanych stron, a pod wieloma względami jest to dziś najlepsza istniejąca encyklopedia. Przed kilkoma laty głośne było porównanie jej z prestiżową *Encyclopædia Britannica* — okazało się, że różnice w poziomie merytorycznym są niewielkie.

O ile przewrót w kategorii encyklopedii właściwie już się dokonał, nieco inaczej wygląda rywalizacja słowników. Oczywiście wyraźnie widać, że i tu papierowe edycje są coraz mniej popularne. Różnice uwidaczniają się, gdy przeanalizowana zostanie sytuacja słowników internetowych. Tak zwany siostrzany projekt Wikipedii, Wikisłownik, nie dominuje wśród

konkurencji — zarówno na świecie, jak i w Polsce. Przyczyny tego stanu rzeczy są złożone. Autor postanowił skupić się na kilku zagadnieniach, uwidaczniających się w polskojęzycznej wersji Wikisłownika. W tym celu konieczne było zbadanie społeczności zaangażowanej w tworzenie tego projektu. Jego efektem było wykonanie prac programistycznych, których opis stanowi główną część niniejszego opracowania.

W przypadku wszystkich projektów opartych na silniku programistycznym MediaWiki istotną barierą rozwoju jest sama technologia. Każdy ochotnik ma możliwość uczestniczenia w rozwoju portalu, wiąże się to jednak z koniecznością przystosowania się do wymagań stawianych przez oprogramowanie. Edytowanie haseł w internetowej encyklopedii czy słowniku jest praktycznie niemożliwe dla osoby bez wcześniejszego przygotowania lub znacznej wiedzy techniczno-informatycznej. Oprogramowanie MediaWiki oparte jest bowiem na tzw. wikikodzie (także: wikitekst, wikiskładnia), czyli języku opisu struktury i wyglądu strony internetowej — prostszym niż HTML, jednak wciąż nieintuicyjnym dla kogoś, kto nie miał wcześniej do czynienia z tego typu edytorami. Dlatego wielu potencjalnych współautorów zniechęca się do projektu już przy pierwszej próbie poprawy artykułu.

Aby zmienić tę sytuację, przygotowany został nowy edytor, dostosowany specjalnie do potrzeb polskiego Wikisłownika. Aplikacja pozwala na o wiele prostsze tworzenie nowych i zmienianie starych haseł niż poprzednia, standardowa. Dzięki użyciu jej jako domyślnej w projekcie popularyzacja edytowania Wikisłownika wśród fachowców w dziedzinach lingwistycznych okaże się łatwiejsze — zniknie podstawowa bariera, jaką jest konieczność dostosowania się do skomplikowanych technicznych wymagań stawianych przez użyte oprogramowanie. Dodatkowo nowa aplikacja umożliwia zaawansowaną automatyzację tworzenia hasła. Wiele z czynności zintegrowanych z nowym edytorem do tej pory wymagało mozolnych poszukiwań w artykułach Wikisłownika oraz innych projektach. Dzięki użyciu API udostępnianego przez serwisy Fundacji Wikimedia skomplikowane przeszukiwanie tysięcy stron udało się sprowadzić do kilku kliknięć.

W dalszej części pracy opisany został proces tworzenia tego edytora. Pierwszy rozdział charakteryzuje pokrótce sam Wikisłownik, jak i pokrewne projekty oraz oprogramowanie w nich użyte. Następnie opisano społecznościowe aspekty tworzenia tego typu aplikacji ze szczególnym uwzględnieniem koncepcji *wiki*. Ostatni rozdział wyczerpująco przedstawia szczegóły projektowe i implementacyjne aplikacji.



## Rozdział 2

# Wikisłownik

Rozdział ten stanowi charakterystykę Wikisłownika — sieciowego słownika opartego na oprogramowaniu MediaWiki. Wikisłownik jest jednym z największych i najpopularniejszych słowników dostępnych w polskim internecie. W kolejnych sekcjach projekt ten został opisany na różnych poziomach szczegółowości. Omówiono zarówno oprogramowanie, na jakim bazuje słownik, jak i swego rodzaju „ekosystem”, w którym znajduje on swoje miejsce.

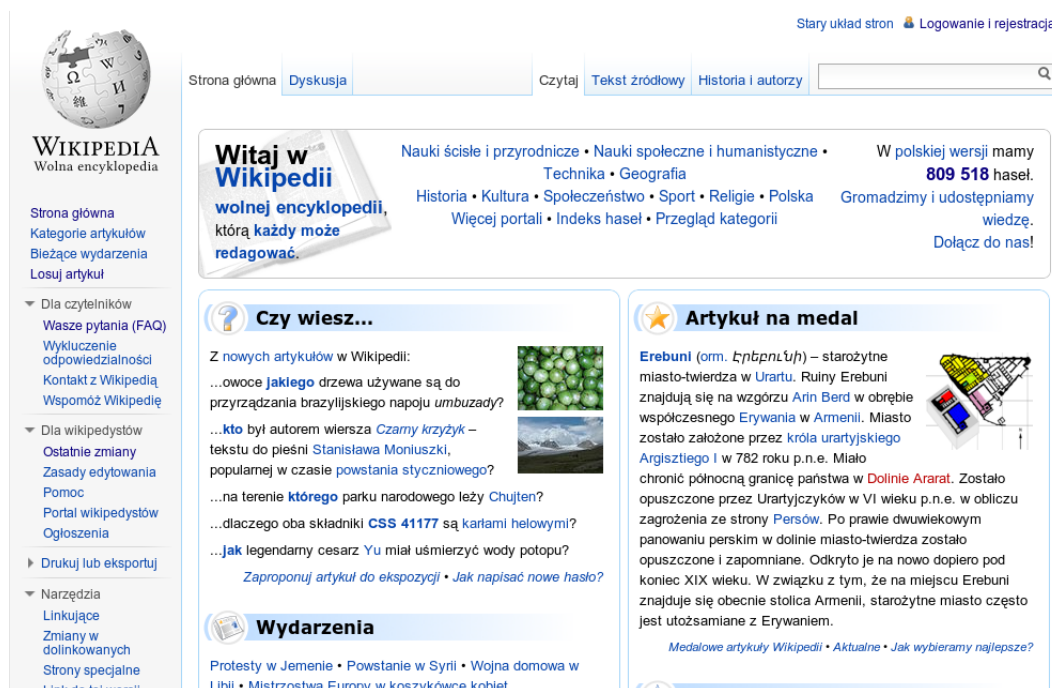
### 2.1. Projekty Fundacji Wikimedia

Podmiotem odpowiedzialnym m.in. za rozwój Wikisłownika jest Wikimedia Foundation Inc. (opisywana dalej jako „Fundacja”) — organizacja non-profit mająca siedzibę w San Francisco w Stanach Zjednoczonych, istniejąca od 2003 roku. Jak informuje strona internetowa polskiego partnera Fundacji, Stowarzyszenia Wikimedia Polska, *celem fundacji jest sprzyjanie tworzeniu i rozwojowi projektów o otwartej treści opartych na technologii WikiWiki oraz dostarczanie społeczności internetowej pełnej zawartości wymienionych projektów za darmo i bez zamieszczania reklam* [1]. Doskonale znaną, sztandarową inicjatywą Fundacji jest Wikipedia (<http://www.wikipedia.org>) — największa obecnie encyklopedia internetowa, dostępna w 281 językach (stan z sierpnia 2011 roku) i zawierająca ponad 19 milionów haseł, w tym ponad 3,7 miliona w największej, angielskojęzycznej<sup>1</sup> edycji [2]. Mimo częstej krytyki tego

---

<sup>1</sup>Oficjalnie w projektach Fundacji używane są określenia typu *angielskojęzyczny*, *polskojęzyczny*. Choć w przypadku wersji polskojęzycznej znakomita większość uczestników projektów pochodzi z Polski, nie jest to regułą dla innych edycji. W dalszej części pracy przyjęto uproszczenie polegające na tym, że określenia typu *polska Wikipedia*, *angielski Wikisłownik* traktowane są jako tożsame z określeniami używającymi sformułowań z częstką

przedsięwzięcia faktem jest, że Wikipedia jest miejscem, z którego miliony osób korzystają, by pozyskać informacje z najróżniejszych dziedzin. Obecny stan rzeczy możliwy jest dzięki pracy wielkiej liczby wolontariuszy tworzących artykuły bez wynagrodzenia.



Ilustracja 2.1: Polska edycja Wikipedii

Wikipedia jest najbardziej znanym, ale nie jedynym projektem pod opieką Fundacji. Pozostałe to tzw. „projekty siostrzane”, w szczególności sposób uwzględniane również przy tworzeniu haseł w encyklopedii. Oto lista wspieranych przez Fundację wielojęzycznych inicjatyw [3]:

- Wikisłownik (ang. *Wiktionary*) — wielojęzyczny słownik internetowy, będący głównym przedmiotem niniejszej pracy,
- Wikicytaty (ang. *Wikiquote*) — zbiór cytatów autorstwa znanych osób, z filmów i książek, przysłów i porzekadeł,
- Wikibooks — serwis z „otwartymi” (opartymi na wolnej licencji) podręcznikami,
- Wikiźródła (ang. *Wikisource*) — zbiór dokumentów źródłowych w wersjach oryginalnych i tłumaczonych, nieograniczonych prawem autorskim,
- Wikinews — otwarty serwis informacyjny,

-języczny.

- Wikiversity — materiały edukacyjne i naukowe,
- Wikispecies — katalog gatunków organizmów żywych,
- MediaWiki — nadzór nad tworzeniem i rozpowszechnianiem oprogramowania MediaWiki (p. sekcja 2.2),
- Meta-Wiki — projekt ułatwiający koordynację wszystkich pozostałych.
- Wikimedia Commons — repozytorium mediów (zdjęć, grafik, filmów) dostępnych na wolnej licencji, z którego korzystają pozostałe projekty Wikimedia,
- Wikimedia Incubator — metaprojekt umożliwiający tworzenie nowych inicjatyw wspieranych przez Fundację.

Wszystkie projekty łączy sposób ich powstawania — możliwość edycji dostępna jest praktycznie dla każdego internauty. Nie dotyczy to co prawda kilku krajów, w których projekty Fundacji zablokowane są w ramach cenzury internetu, jednak ogromna większość osób dysponujących łączem internetowym ma szansę stać się jednymi spośród współautorów haseł.

Drugą cechą wspólną są wolne licencje, na których udostępniana jest zawartość wszystkich serwisów. Po reformie w czerwcu 2009 roku treść Wikipedii i projektów siostrzanych dostępna jest nie tylko na licencji GNU FDL (Free Documentation License) 1.2, ale także na kompatybilnej z nią CC-BY-SA 3.0 (Creative Commons Attribution-ShareAlike / Uznanie Autorstwa — Na Tych Samych Warunkach) [4]. Oznacza to, że można ją dowolnie wykorzystywać we własnych dziełach pod warunkiem podania oryginalnych autorów i zachowania pierwotnej licencji.

## 2.2. Oprogramowanie MediaWiki

Sama działalność wolontariacka redaktorów projektów Wikimedia nie wystarczyłaby do stworzenia serwisów internetowych o obecnych kształtach. Konieczne jest oczywiście również zapewnienie oprogramowania, które umożliwi płynną współpracę przy tworzeniu haseł. Tym oprogramowaniem jest wolna platforma MediaWiki stworzona zgodnie z zasadami *open source*. System MediaWiki napisany jest w języku PHP i obsługuje kilka popularnych baz danych (w przypadku projektów Wikimedia jest to MySQL). Dla inicjatyw Wikimedia

stanowi szkielet programistyczny od samego ich początku, a od 2002 roku stale się rozwija. W czerwcu 2011 roku wersją używaną w projektach było MediaWiki 1.17.

System MediaWiki używany jest nie tylko w projektach wspieranych przez Fundację, ale także w tysiącach innych, mniejszych lub większych, co jest możliwe dzięki wysokiemu stopniowi konfigurowalności i dużej liczbie dostępnych rozszerzeń. Są to w dużej mierze serwisy o podobnym charakterze, umożliwiające swobodną wymianę informacji na dowolny temat. MediaWiki bywa także używane w firmowych intranetach i wszędzie tam, gdzie zachodzi potrzeba udostępnienia materiałów do edycji dużej liczbie użytkowników.

### **2.2.1. Edytowanie i wikitekst**

Strony w projektach opartych na platformie MediaWiki na ogół nie mogą być czystym tekstem, pozbawionym formatowania. Przykładowo hasła w encyklopedii muszą zachowywać określoną strukturę — występuje więc podział na sekcje, ilustracje, różne rodzaje formatowania (kursywa, wytłuszczenie), przypisy czy powtarzalne fragmenty. Szczególnie istotnym elementem są linki pomiędzy poszczególnymi artykułami, wyróżniające projekty Fundacji na tle ich papierowych, ale też elektronicznych konkurentów. Odnośniki pozwalają błyskawicznie przemieszczać się między hasłami, by w ten sposób uzyskiwać kolejne informacje wspomagające przyswajanie wiedzy.

Linki i formatowanie na stronach internetowych tworzone są za pomocą elementów języka HTML lub XHTML. O ile języki te są proste w obsłudze dla specjalisty informatyka, to laik nie jest w stanie stworzyć za ich pomocą stron bez uprzedniego dłuższego przygotowania. Aby umożliwić bezproblemową edycję stron internetowych osobom bez wykształcenia informatycznego, programiści MediaWiki zaprojektowali tzw. wikitekst [5] — uproszczony język opisu stron, pozwalający na realizację wymienionych elementów. Porównanie niektórych z nich znajduje się w tabeli 2.1.

Łatwo można zauważyć, że używanie wikitekstu jest o wiele prostsze niż nauka języka XHTML. Jeśli zachodzi potrzeba zaawansowanego formatowania, możliwe jest także użycie znaczników XHTML. W przypadku standardowego formatowania jest to jednak niewskazane ze względu na dobro niedoświadczonych edytorów.

Bardzo istotnym elementem wikitekstu są szablony — predefiniowane fragmenty kodu z opcjonalnymi parametrami. Szablony można uznać za odpowiednik procedur/funkcji

	<b>Wikitekst</b>	<b>XHTML</b>
Kursywa	<code>''Tekst''</code>	<code>&lt;em&gt;Tekst&lt;/em&gt;</code>
Wytłuszczenie	<code>'''Tekst'''</code>	<code>&lt;strong&gt;Tekst&lt;/strong&gt;</code>
Nagłówek	<code>== Nagłówek ==</code> <code>=== Nagłówek ===</code>	<code>&lt;h2&gt;Nagłówek&lt;/h2&gt;</code> <code>&lt;h3&gt;Nagłówek&lt;/h3&gt;</code>
Odnośnik wewnętrzny	<code>[[Strona]]</code> <code>[[Strona strony]]</code>	<code>&lt;a href="/wiki/Strona"&gt;Strona&lt;/a&gt;</code> <code>&lt;a href="/wiki/Strona"&gt;strony&lt;/a&gt;</code>
Odnośnik zewnętrzny	<code>[http://www.google.com Google]</code>	<code>&lt;a href="http://www.google.com"&gt;Google&lt;/a&gt;</code>
Obraz	<code>[[Plik:Przykład.png thumb Podpis]]</code>	<code>&lt;img src="../../../Przykład.png"/&gt;&lt;br/&gt;</code> <code>&lt;div class="caption"&gt;Podpis&lt;/div&gt;</code>
Podział na akapity	Pierwszy akapit  Drugi akapit	<code>&lt;p&gt;Pierwszy akapit&lt;/p&gt;</code> <code>&lt;p&gt;Drugi akapit&lt;/p&gt;</code>
Lista nienumerowana	<code>* Element</code> <code>* Element</code> <code>* Element</code>	<code>&lt;ul&gt;</code> <code>&lt;li&gt;Element&lt;/li&gt;</code> <code>&lt;li&gt;Element&lt;/li&gt;</code> <code>&lt;li&gt;Element&lt;/li&gt;</code> <code>&lt;/ul&gt;</code>
Lista numerowana	<code># Element</code> <code># Element</code> <code># Element</code>	<code>&lt;ol&gt;</code> <code>&lt;li&gt;Element&lt;/li&gt;</code> <code>&lt;li&gt;Element&lt;/li&gt;</code> <code>&lt;li&gt;Element&lt;/li&gt;</code> <code>&lt;/ol&gt;</code>

Tabela 2.1: Porównanie HTML i wikitekstu

w językach programowania. W projektach opartych na MediaWiki szablony pełnią przede wszystkim dwie główne funkcje:

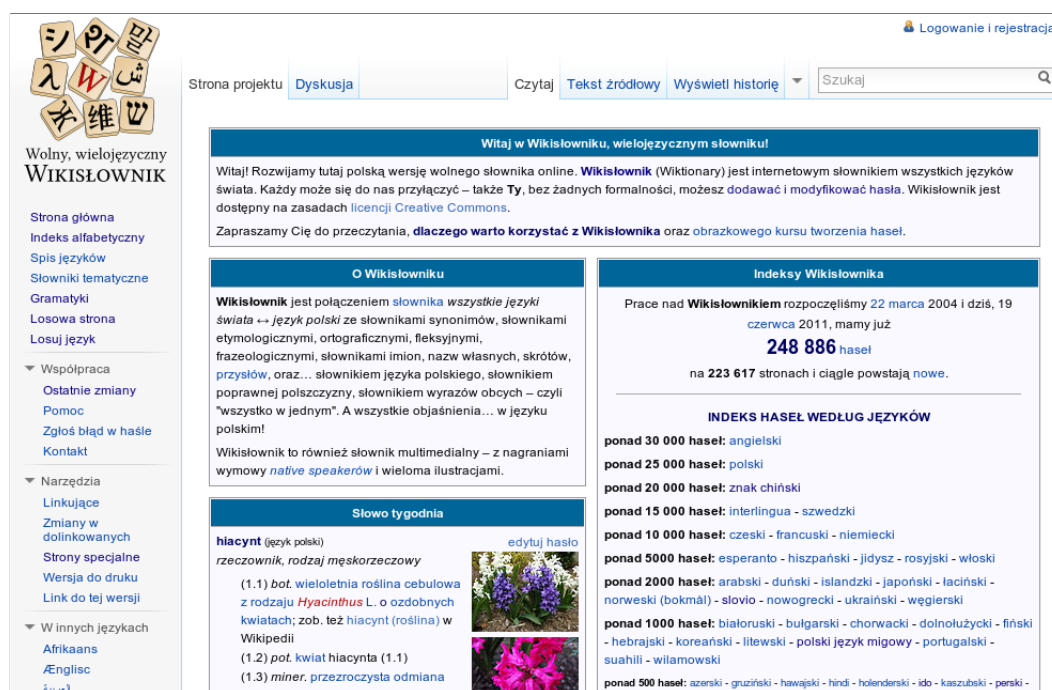
- upraszczają kod — pozwalają np. na zastąpienie skomplikowanego kodu XHTML (a także jeszcze bardziej złożonych funkcji parsera MediaWiki) krótkim wywołaniem szablonu,
- standaryzują strony — często wykorzystywane fragmenty wywoływane są zawsze w dokładnie ten sam sposób.

We wszystkich większych projektach Fundacji szablony są bardzo często wykorzystywane. Przy tym stanowią duże ułatwienie dla technicznie zaawansowanych autorów, którzy wspomagają się przy tworzeniu haseł dodatkowymi technologiami. Na używaniu szablonów korzystają przede wszystkim boty, czyli programy dokonujące edycji samodzielnie po uprzednim przygotowaniu lub pod stałą opieką programisty. W większości projektów przyjęte jest, że każdy bot ma własne konto użytkownika, nie używa natomiast konta swojego „właściciela”. Dzięki botom możliwe jest np. masowe tworzenie haseł w Wikipedii na

ściśle określony temat, jeśli istnieje dobre źródło w formie czytelnej dla komputera (takich jak opisy asteroid czy wszystkich miejscowości lub jednostek administracyjnych w danym kraju). Innym ich zastosowaniem jest automatycznie uzupełnianie tzw. interwiki — czyli odnośników pomiędzy poszczególnymi wersjami językowymi tego samego hasła.

Szablony mogą być wykorzystywane w prosty sposób nawet przez początkujących użytkowników — aby wywołać szablon, wystarczy wpisać jego nazwę pomiędzy podwójnymi nawiasami klamrowymi ({{Nazwa szablonu}}) lub ({{Nazwa szablonu|parametr=abc}}). W polskim Wikisłowniku szablony pełnią szczególną rolę — i to m.in. dzięki ich szerokiemu zastosowaniu w projekcie zrodził się pomysł na niniejszą pracę. Szczegóły tego zagadnienia zostaną przedstawione w sekcji 2.4.

## 2.3. Wiktionary — Wikisłownik



Ilustracja 2.2: Polska edycja Wikisłownika

Jednym z największych projektów siostrzanych Wikipedii jest Wikisłownik, w wersji angielskiej (i wielu innych) noszący nazwę *Wiktionary* (<http://www.wiktionary.org>). Ten słownik internetowy nie rozwinął się jeszcze tak prędko jak encyklopedia, zwłaszcza jeśli chodzi o polską edycję. Jest dziś jednak jednym z największych słowników w sieci, a w pewnych

zastosowaniach stanowi najlepszy wybór. Dużą zaletą Wikisłownika jest jego wielojęzyczność — w tym samym serwisie znaleźć można hasła w ponad 250 językach. W przypadku niektórych z nich jest to praktycznie jedyny słownik internetowy lub nawet jedyny dostępny słownik w ogóle. Przykładem może być polski Wikisłownik, który zawiera prawdopodobnie jedyny polski słownik języka hawajskiego czy największe słowniki języków suahili i jidysz [6].

Wspomniane zostało zastosowanie szablonów do standaryzacji kodu źródłowego i struktury haseł. Trzeba jednak zaznaczyć, że dotyczy to wyłącznie haseł w obrębie jednej wersji językowej Wikisłownika, są one bowiem niezależne od siebie i od Fundacji. Wspólne dla wszystkich edycji jest jedynie oprogramowanie MediaWiki i umiejscowienie na serwerach Fundacji pod adresem `xxx.wiktionary.org`, gdzie zamiast `xxx` wstawiany jest dwu- lub trzyliterowy skrót ISO języka (np. `pl.wiktionary.org` = język polski, `de.wiktionary.org` = język niemiecki, `sq.wiktionary.org` = język albański, `csb.wiktionary.org` = język kaszubski). Wszystkie kwestie organizacyjne w obrębie wersji językowej ustalane są w ramach dyskusji i głosowań przez internetową społeczność. Głosowania służą także wyborowi administratorów projektu, czyli użytkowników mających dodatkowe uprawnienia, spośród których najważniejsze to usuwanie i zabezpieczanie haseł oraz blokowanie użytkowników działających na szkodę projektu. W lipcu 2011 roku w angielskim Wikisłowniku działało aktywnie 76 administratorów [7], zaś w polskim — 14 [8]. Dla porównania Wikipedia w języku angielskim ma 1541 administratorów (niekoniecznie aktywnych), wersja polska natomiast 163 [9].

Podobnie jak w przypadku Wikipedii, hasła w poszczególnych wersjach językowych Wikisłownika są łączone poprzez odnośniki interwiki, znajdujące się na dole lewego menu w większości haseł. W stosunku do encyklopedii występuje znacząca różnica w sposobie funkcjonowania tych linków. Wikipedia poprzez interwiki łączy artykuły na ten sam temat, często różniące się tytułem (polski artykuł *Kot domowy* odsyła do angielskiego *Cat* czy niemieckiego *Hauskatze*). W Wikisłowniku mechanizm interwiki łączy zaś strony o tym samym tytule, niezależnie od ich zawartości. Polskie hasło *kot* zawiera więc łącza do haseł o tytule *kot* w innych językach, które mogą, ale nie muszą zawierać m.in. objaśnienia polskiego znaczenia tego słowa. Z tego względu łatwo jest odnaleźć brakujące informacje o wybranym słowie w danym języku, jeśli skorzysta się z kilku edycji językowych Wikisłownika. Natomiast tłumaczenia tytułu hasła na inne języki znajdują się bezpośrednio w treści artykułu,

w miejscu przeznaczonym dla takich informacji.

Choć poszczególne wersje Wikisłownika różnią się między sobą, wspólna jest struktura hasła na najogólniejszym poziomie. Każde hasło jest podzielone na sekcje nagłówkami 2. stopnia (p. tabela 2.1), podobnie jak w innych projektach Fundacji. W przypadku słownika sposób podziału artykułu został jasno sprecyzowany i jest wspólny dla wszystkich edycji. W każdej z sekcji objaśniono znaczenie tytułowego hasła w innym języku. Przykładowo hasło *nie* w większości wersji językowych ma sekcje objaśniające znaczenie m.in. w języku polskim oraz języku niemieckim (*nigdy*). Właśnie ta ogólna cecha struktury haseł pozwala na częściową automatyzację niektórych często wykonywanych w trakcie edycji czynności.

Aplikacja opisana w niniejszej pracy przeznaczona jest dla polskiej edycji Wikisłownika. Konieczne jest zatem scharakteryzowanie specyfiki tego projektu (p. sekcja 2.4). Ze względu na odmiennność poszczególnych wersji językowych prawdopodobnie w innych nie będzie można wykorzystać aplikacji. Z pewnością może ona być dla nich jednak bardzo przydatna — duża część funkcji przez nią realizowanych może być używana niezależnie od specyfiki danej wersji, a budowa aplikacji pozwoli na ich wyodrębnienie.

## 2.4. Polska edycja Wikisłownika

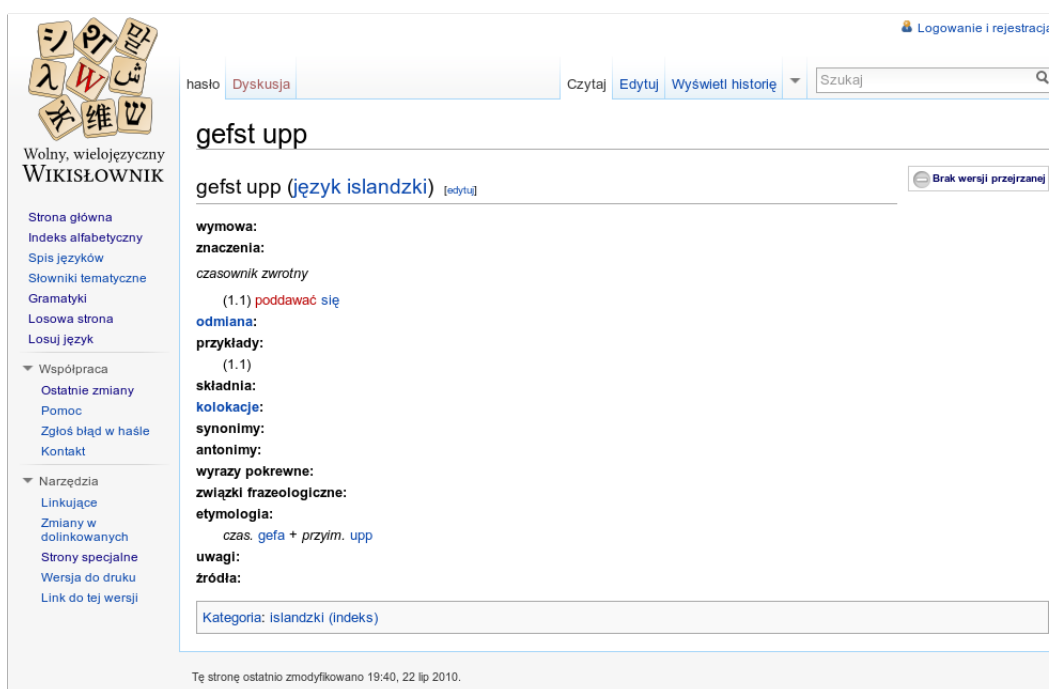
Wikisłownik w języku polskim powstał 22 marca 2004 r. [10] i jest dziś jedną z najlepiej rozwiniętych edycji. Jeśli chodzi o liczbę haseł, polska wersja zawiera ich ponad 230 000 i znajduje się na 8. miejscu (pierwsze trzy miejsca zajmują z dużą przewagą edycje angielska, francuska i chińska). Warto zwrócić jednak uwagę, że liczba artykułów nie musi odzwierciedlać ogólnego poziomu rozwoju danej edycji, a przynajmniej w o wiele mniejszym stopniu niż w Wikipedii. Automatyczne importowanie haseł słownikowych z innych źródeł jest prostsze niż w przypadku wpisów do encyklopedii, a hasła takie oczywiście nie wyróżniają się pozytywnie pod względem jakości. Wydaje się, że lepszym wskaźnikiem rozwoju są np. liczby aktywnych użytkowników lub administratorów. Tutaj polski Wikisłownik okazuje się wielokrotnie lepszy niż edycje litewska czy malajska, górujące nad nim pod względem liczby haseł [11].

W kolejnej podsekcji opisana została dokładnie struktura haseł w polskim Wikisłowniku. Społeczność tego projektu scharakteryzowano natomiast w sekcji 3.2.



### 2.4.1. Struktura hasła

Jak wspomniano, wspólny dla wszystkich wersji językowych Wikisłownika jest podział na sekcje, odpowiadające poszczególnym językom. Polska edycja Wikisłownika jest bardzo dobrze ustrukturyzowana w dalszym zakresie, co niekoniecznie jest regułą dla pozostałych edycji. Wynikiem tego jest jednolity wygląd hasła, który otrzymuje czytelnik, oraz jednolity kod widziany przez redaktora. Przykładowy artykuł z jedną sekcją językową znajduje się na ilustracji 2.3.

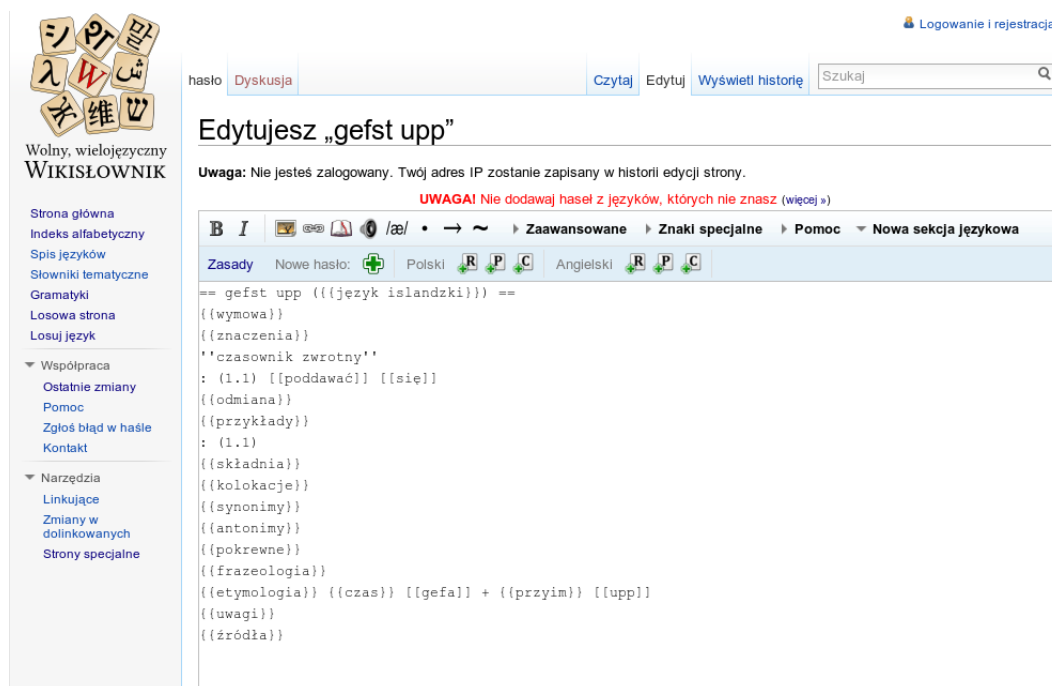


Ilustracja 2.3: Hasło *gefst upp* w polskim Wikisłowniku ([http://pl.wiktionary.org/wiki/gefst\\_upp](http://pl.wiktionary.org/wiki/gefst_upp))

Na rzucie ekranu widoczny jest układ sekcji typowy dla polskiego projektu. Z małymi wyjątkami sekcje danego języka wyglądają tak samo we wszystkich hasłach. To oznacza, że także w innych opisanych wyrazach języka islandzkiego znaleźć można elementy: wymowa, znaczenia, odmiana, przykłady, składnia, kolokacje, synonimy, antonimy, wyrazy pokrewne, związki frazeologiczne, etymologia, uwagi, źródła. W rzeczywistości jest to układ typowy dla znakomitej większości występujących w Wikisłowniku języków [12]. Odstępstwa od tego szkieletu hasła są niewielkie i zostaną przedstawione poniżej.

Ilustracja 2.4 przedstawia wygląd przykładowej strony po przejściu do trybu edycji, dostępnego dla każdego użytkownika (bez konieczności logowania). Widok ten prezentuje

źródłowy wikikod hasła, dodatkowo edytor ma kilka ikon ułatwiających wstawianie często używanych elementów. W kodzie tym widoczne są szablony definiujące tytuły poszczególnych elementów sekcji (nazywanych dalej *podsekcjami*). Podział na sekcje i podsekcje jest dość prosty do wykonania za pomocą programu komputerowego i już sama ta procedura pozwala na znaczne zwiększenie przejrzystości w nowym edytorze, opisanym w rozdziale 4.



Ilustracja 2.4: Edycja hasła *gefst upp* w polskim Wikisłowniku ([http://pl.wiktionary.org/w/index.php?title=gefst\\_upp&action=edit](http://pl.wiktionary.org/w/index.php?title=gefst_upp&action=edit))

Każda z podsekcji w hasłach ma ściśle określoną funkcję, dzięki której polski Wikisłownik jest jednolity. Poniżej zostały omówione wszystkie podsekcje występujące w artykułach, w jedynej przyjętej jako poprawną kolejności. Duża ich część pojawia się jedynie w niektórych językach, dlatego choć lista podsekcji jest dość długa, to każde hasło będzie miało ich o wiele mniej. Szablony podsekcji zebrane są na stronie [http://pl.wiktionary.org/wiki/Kategoria:Szablony\\_szablonów\\_haseł](http://pl.wiktionary.org/wiki/Kategoria:Szablony_szablonów_haseł).

## Podsekcje występujące w hasłach polskiego Wikisłownika

**Szablon**      `{{zapis hieroglificzny}}`

**Zawartość**      Zapis hieroglificzny słowa w języku staroegipskim, pokazany za pomocą grafik PNG z repozytorium Wikimedia Commons. Oznaczenie (1.1) odnosi

	się do numeracji w sekcji <i>znaczenia</i> .
<b>Języki</b>	tylko staroegipski <sup>2</sup> .
<b>Przykład</b>	<pre>{{zapis hieroglificzny}}</pre> <pre>: (1.1) [[Plik:Egyptian-Pr-cnḥ.PNG]];</pre> <pre>[[Plik:Egyptian-Pr-cnḥ2.PNG]];</pre> <pre>[[Plik:Egyptian-Pr-cnḥ3.PNG]]</pre> <p style="text-align: center;">*</p>
<b>Szablon</b>	<pre>{{ortografie}}</pre>
<b>Zawartość</b>	Inne sposoby zapisu tytułu hasła. Zazwyczaj chodzi o alternatywną pisownię w języku, do zapisu którego używane są dwa alfabety (np. serbski). Do prezentacji pisowni mogą być używane szablony wyświetlające dodatkowe informacje.
<b>Języki</b>	azerski, białoruski, dżuhuri, gagauski, krymskotatarski, ladino, serbski, slovio, tatarski, turkmeński, ujgurski
<b>Przykład</b>	<pre>{{ortografie}}</pre> Македониа
	*
<b>Szablon</b>	<pre>{{transliteracja}}</pre>
<b>Zawartość</b>	Transliteracja słowa zapisanego w obcym alfabecie na alfabet łaćniński. W przeciwieństwie do transkrypcji transliteracja może być wykonywana automatycznie — każda litera alfabetu obcego konwertowana jest na jeden lub więcej znaków w alfabecie łaćnińskim. Obecnie często używany jest szablon <pre>{{translit}}</pre> , który umożliwia automatyczną konwersję za pomocą JavaScriptu niektórych alfabetów podczas odczytywania strony.
<b>Języki</b>	abazyński, abchaski, adygejski, akadyjski, amharski, arabski, aramejski, asamski, awarski, baszkirski, beludzi, bengali, birmański, bułgarski, chaka-ski, czeczeński, czuwaski, dzongkha, erzja, gocki, gruziński, gudżarati, gyz, hebrajski, hindi, inguski, inuktitut, jidysz, kannada, kaszmirski, kazachski, khmerski, kirgiski, komi, komi-jażwiński, kri, kurdyjski, laotański, lezgiński, macedoński, malajalam, malediowski, marathi, maryjski, mongol-

<sup>2</sup>Rubryka *Języki* precyzuje, w których sekcjach językowych występuje dana podsekcja. Oprócz zwykłych sekcji, odpowiadających danemu językowi, istnieją także nietypowe: *użycie słowa obcego w języku polskim* i *znak chiński*

ski, nepalski, newarski, nowogrecki, orija, ormiański, osetyjski, paszto, pendżabski, perski, romski, rosyjski, sanskryt, sindhi, sorani, staro-cerkiewno-słowiański, starogrecki, staroormiański, sumeryjski, syngaleski, tabasarański, tadżycki, tajski, tamazight, tamilski, telugu, tybetański, ukraiński, urdu, zarfatit

**Przykład** `{{transliteracja}}` Moskva

\*

**Szablon** `{{transkrypcja}}`

**Zawartość** Transkrypcja słowa zapisanego w obcym alfabecie na język polski, czyli przedstawienie go w formie dającej informacje o rzeczywistej wymowie

**Języki** standardowo tylko staroegipski, podsekcja ta może być jednak dodawana w wielu innych językach (przykład w jidysz)

**Przykład** `{{transkrypcja}}`  
: (1.1-3) `{{YIVO|{{lp}} khaver {{lm}} khaveyrim}}`; polska: `{{lp}}`  
chawer `{{lm}}` chawejrim  
: (1.4) `{{YIVO|{{lp}} khover {{lm}} khovers}}`; polska: `{{lp}}`  
chower; `{{lm}}` chowers

\*

**Szablon** `{{czytania}}`

**Zawartość** Wyjaśnienie możliwego wymawiania znaków kanji używanych w języku japońskim. Występują dwa sposoby czytania: on’yomi i kun’yomi. Do ich prezentacji używane są szablony `{{on}}` i `{{kun}}`.

**Języki** tylko japoński

**Przykład** `{{czytania}}` `{{on}}` ビ (bi); `{{kun}}` はな (hana)

\*

**Szablon** `{{klucz}}`

**Zawartość** Elementy, według których układane są słowniki języka chińskiego.

**Języki** tylko znak chiński

**Przykład** `{{klucz}}` 157 足 + 6

\*

<b>Szablon</b>	<code>{{kreski}}</code>
<b>Zawartość</b>	Liczba kresek użytych do napisania danego znaku. Informacja służy m.in. do ułatwienia odnajdywania znaków w papierowych słownikach.
<b>Języki</b>	znak chiński, koreański
<b>Przykład</b>	<code>{{kreski}}</code> 13

\*

<b>Szablon</b>	<code>{{warianty}}</code>
<b>Zawartość</b>	Szablon powiększający znak chiński i ewentualnie jego warianty. Szablon ten jest używany inaczej niż większość: nie odpowiada jedynie za wyświetlenie nagłówka, a zawartość sekcji wstawiana jest jako parametr. Najczęściej w parametrze pojawia się szablon <code>{{zch-w}}</code> , odpowiadający za prezentację znaku.
<b>Języki</b>	tylko znak chiński
<b>Przykład</b>	<code>{{warianty {{zch-w}}}}</code>

\*

<b>Szablon</b>	<code>{{kolejność}}</code>
<b>Zawartość</b>	Kolejność stawiania kresek w znaku chińskim, ilustrowana za pomocą grafiki dostępnej w projekcie Wikimedia Commons. W tej podsekcji używane są szablony <code>{{zch-komiks}}</code> , <code>{{zch-cienie}}</code> i <code>{{zch-animacja}}</code> , które ładują automatycznie grafiki o nazwie odpowiadającej hasłu, jeśli te istnieją.
<b>Języki</b>	tylko znak chiński
<b>Przykład</b>	<code>{{kolejność}}</code> <code>{{zch-komiks}}</code>

\*

<b>Szablon</b>	<code>{{wymowa}}</code>
<b>Zawartość</b>	Jedna z kluczowych podsekcji we wszystkich językach. Podawane są w niej informacje na temat wymowy danego hasła, zarówno za pomocą alfabetów fonetycznych (jak np. IPA oraz alfabet słowiański), jak i nagrań dźwiękowych umieszczonych w Wikimedia Commons. Do opisywania wymowy stosowane są szablony <code>{{IPA}}</code> , <code>{{IPA2}}</code> , <code>{{IPA3}}</code> , <code>{{IPA4}}</code> . Wymowa słów

polskich dodawana jest automatycznie przez bota uruchomionego przez jednego z administratorów Wikisłownika. Bot ten jest skomplikowanym programem napisanym w Javie [13], generującym wymowę na podstawie publikacji Danuty Ostaszewskiej i Jolanty Tambor [14]. Pliki dźwiękowe dodawane są szablonem `{{audio}}`.

## Języki

wszystkie poza znakiem chińskim i użyciem międzynarodowym

## Przykład

```
{{wymowa}} {{audio|Pl-samochód.ogg}}, {{IPA3|să'mɔxut}},
{{AS3|sām''o''χut}}, {{objaśnienie wymowy|WYG|NAZAL}}
```

\*

## Szablon

`{{znaczenia}}`

## Zawartość

Jedyna podsekcja obowiązkowa, w której podawane jest znaczenie hasła. Zawartość podsekcji dzielona jest najpierw na części mowy, potem na poszczególne znaczenia, które zostają ponumerowane zgodnie z obowiązującym schematem. W hasłach polskich podawane jest dłuższe znaczenie, w innych językach tłumaczenie na polski. Znaczenia te są linkami do objaśnień form podstawowych poszczególnych słów — linkowanie to stanowi dość duże utrudnienie przy tworzeniu hasła.

## Języki

wszystkie

## Przykłady

Hasło angielskie:

```
{{znaczenia}}
''rzeczownik''
: (1.1) [[zamówienie]]
: (1.2) [[rozkaz]]
: (1.3) [[porządek]]
: (1.4) {{syst}} [[rząd]]
: (1.5) {{mat}} [[rząd]]
''czasownik''
: (2.1) [[zamawiać]]
: (2.2) [[rozkazywać]]
```

Hasło polskie:

```
{{znaczenia}}
```

```

''rzeczownik, rodzaj męski''
: (1.1) [[budynek]] [[warowny]]; {{wikipedia|zamek (architektura)}}
: (1.2) [[mechanizm]] [[zamykać|zamykający]] [[drzwi]],
[[szuflada|szuflady]]; {{wikipedia|zamek (urządzenie)}}
: (1.3) [[zapięcie]] [[garderoba|garderoby]],
{{zob|[[zamek błyskawiczny]]}}.
: (1.4) [[element]] [[składowy]] [[broń|broni]] [[palny|palnej]];
{{wikipedia|zamek (broń)}}
: (1.5) {{sport}} [[w]] [[hokej]]u: [[zamykać|zamknięcie]]
[[przeciwnik]]a [[w]] [[tercja|tercji]], [[gdy]] [[drużyna]]
[[atakować|atakująca]] [[rozgrywać|rozgrywa]] [[krążek]] [[w]]
[[tercja|tercji]] [[przeciwnik]]a [[nie]] [[pozwalać|pozwalając]]
[[on|mu]] [[wyjść]] [[poza]] [[niebieski|niebieską]] [[linia|linię]]

```

\*

<b>Szablon</b>	{{determinatywy}}
<b>Zawartość</b>	Znak określający, o jaką klasę znaczeniową wyrazów chodzi w danym haśle. Wyświetlana jest grafika z Wikimedia Commons.
<b>Języki</b>	tylko staroegipski
<b>Przykład</b>	{{determinatywy}} <p>: (1.1) [[Plik:Egyptian-nb 'nh-determinative.PNG]]</p>

\*

<b>Szablon</b>	{{odmiana}}
<b>Zawartość</b>	Odmiana wyrazu, prezentowana na różne sposoby. Występują np. szablony, które generują odmianę na podstawie formy podstawowej hasła. Niekiedy podawana jest jedynie deklinacja lub koniugacja (z odnośnikiem do tabel ją przedstawiających), gdzie indziej cała odmiana.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{odmiana}} <p>: (1.1-2) читать {{ter}} {{lp}} читаю, читаешь, читает; {{lm}} читаем, читаете, читают; {{przesz}} {{lp}} читал / читала / читало; {{lm}} читали; {{rozk}} {{lp}} читай; {{ims}} читающий; читаемый; читая</p>

\*

<b>Szablon</b>	{{przykłady}}
<b>Zawartość</b>	Przykłady użycia danego słowa w zdaniu. W przypadku języków innych niż polski tłumaczenie przykładu podane jest po znaku →. Podobnie jak znaczenia, przykłady są linkowane.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{przykłady}} : (2.1) ''[[Michael]] ''locks'' [[his]] [[house]] [[every]] [[day]]. '' → [[Michał]] [[codziennie]] ''[[zamykać zamyka]]'' [[swój]] [[dom]] (na klucz).

\*

<b>Szablon</b>	{{składnia}}
<b>Zawartość</b>	Podsekcja zawiera informacje o używaniu słowa w połączeniu z przyimkami czy przypadkami.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{składnia}} : (1.2) jechać +{{N}}; jechać [[do]] +{{D}}, jechać [[na]] +{{B}}

\*

<b>Szablon</b>	{{kolokacje}}
<b>Zawartość</b>	Kolokacje to często używane zestawienia słów, w których (w przeciwieństwie do związków frazeologicznych) znaczenie całości wynika ze znaczenia poszczególnych wyrazów.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{kolokacje}} [[mieć]] / [[budzić]] / [[odbierać]] nadzieję • [[promyk]] nadziei • [[ziścić]] nadzieje • [[karmić]] [[się]] nadzieją

\*

<b>Szablon</b>	{{synonimy}}
<b>Zawartość</b>	Wyrazy bliskoznaczne, synonimy.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{synonimy}} : (1.1) [[ufność]], [[wiara]], [[zawierzenie]], [[pociecha]]



\*

<b>Szablon</b>	{{antonimy}}
<b>Zawartość</b>	Wyrazy przeciwstawne, antonimy
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{antonimy}} : (1.1) [[rezygnacja]], [[zwątpienie]], [[beznadzieja]]

\*

<b>Szablon</b>	{{złożenia}}
<b>Zawartość</b>	W językach japońskim i koreańskim podsekcja ta podaje słowa, które powstają jako złożenie danego hasła z innym. Użyty w przykładzie szablon {{furi}} pomaga dobrze wyświetlić furiganę — japońskie pismo.
<b>Języki</b>	koreański i japoński
<b>Przykład</b>	{{złożenia}} {{furi  五日   いつか}}, {{furi  五月   ごがつ}}, {{furi  五輪   ごりん}}, {{furi  五輪大会   ごりんたいかい}}

\*

<b>Szablon</b>	{{pokrewne}}
<b>Zawartość</b>	Wyrazy pokrewne do danego wyrazu podstawowego. W przypadku większej grupy wyrazów wspólnej dla wielu haseł może wystąpić odsyłacz do wyrazu podstawowego, np. w hasle <i>kocur</i> : {{zob [[kot]]}}.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	{{pokrewne}} : (1.1) {{rzecz}} [[picklock]], [[locksmith]], [[locknut]] : (1.2) {{rzecz}} [[dreadlock]] : (1.3) {{rzecz}} [[airlock]], [[lockage]] : (2.1) {{przym}} [[lockable]]; {{rzecz}} [[locker]]

\*

<b>Szablon</b>	{{pochodne}}
<b>Zawartość</b>	Odpowiednik podsekcji <i>pokrewne</i> dla morfemów w esperanto, stanowiących osobne hasła.
<b>Języki</b>	esperanto
<b>Przykład</b>	{{pochodne}} {{rzecz}} [[zebro]], [[zebrino]], [[zebrido]]

\*

<b>Szablon</b>	{{frazologia}}
<b>Zawartość</b>	Podsekcja zawiera związki frazeologiczne, które prezentowane są podobnie jak kolokacje. Różnica między kolokacjami a związkami frazeologicznymi polega na tym, że w przypadku tych drugich znaczenie związku nie wynika bezpośrednio ze znaczeń poszczególnych wyrazów.
<b>Języki</b>	wszystkie poza znakiem chińskim
<b>Przykład</b>	<div>{{frazologia}}</div> <div>: [[psi urok]] • [[tu leży pies pogrzebany]] • {{wulg}} [[pies kogoś jebał]] • [[pies ogrodnika]] • {{pot}} [[pies na baby]] • [[pogoda pod psem]] • [[psu na budę]] • [[pieskie życie]] • [[psia wachta]] • [[psia koja]] • [[pies Pawłowa]] • [[nie dla psa kiełbasa]] • [[pies łańcuchowy Darwina]] • [[na psa urok]] • [[ni pies, ni wydra]] • [[schodzić na psy]] • [[łgać jak pies]] • [[delikatny jak francuski piesek]] • [[francuski piesek]] • [[pies z nim tańczył]] • [[psi żywot]] • [[psie figle]] • [[psi obowiązek]] • [[całować psa w nos]] • [[wyć jak pies do księżycy]] • [[żyć jak pies z kotem]] • [[wieszać na kimś psy]] • [[lubić kogoś jak psy dziada w ciasnej ulicy]] • [[robić coś psim śwędem]] • [[wyglądać jak zbity pies]] • [[być wyszczekany jak pies]] • [[psia kość]] • [[kupować za psie pieniądze]] • [[wyszczeakać coś jak pies]] • [[odszczekać coś jak pies]] • [[wierny jak pies]] • [[pies ci mordę lizał]]</div> <div>: zobacz też: [[Aneks:Przysłowia polskie - zwierzęta#pies przysłowia o psie]]</div>

\*

<b>Szablon</b>	{{etymologia}}
<b>Zawartość</b>	Pochodzenie wyrazu, zapisywane za pomocą szablonów {{etym}} i {{etymn}}.
<b>Języki</b>	wszystkie
<b>Przykład</b>	<div>{{etymologia}}</div> <div>: {{etym prasłowiański *ne}} &lt; {{etym praindoeuropejski *ne}} 'nie'</div> <div>: {{por}} {{etymn czeski ne}}, {{etymn rosyjski ne}},</div>

{{etymn|litewski|ne}}, {{etymn|łaciński|ne}}

\*

**Szablon**      {{kody}}

**Zawartość**    Informacje na temat wprowadzania znaków chińskich za pomocą klawiatury w różnych metodach oraz kodowania Unicode. Podobnie jak w przypadku podsekcji *warianty*, i tutaj szablon przyjmuje parametry pozwalające na zestandaryzowane wyświetlanie.

**Języki**        tylko znak chiński

**Przykład**     {{kody |cjz= 田金 |cjl=WC |cr=6021<sub>0</sub> |u=56db}}

\*

**Szablon**        {{hanja}}

**Zawartość**    W tej podsekcji podawana jest pisownia danego słowa koreańskiego w piśmie hanja (hancha), czyli pisownia zapożyczona z języka chińskiego. Częściej w koreańskim używany jest alfabet hangul.

**Języki**        tylko koreański

**Przykład**     {{hanja}} [[憲法]]

\*

**Szablon**        {{słowniki}}

**Zawartość**    Informacja na temat występowania danego znaku chińskiego w słownikach KangXi, Dai Kanwa Jiten, Dae Jaweon i Hanyu Da Zidian.

**Języki**        tylko znak chiński

**Przykład**     {{słowniki|kx=1163.080|dkj=35533|dj=1628.020|hdz=63974.090}}

\*

**Szablon**        {{uwagi}}

**Zawartość**    Dodatkowe informacje, np. częste błędy, odpowiedzi na typowe wątpliwości.

**Języki**        wszystkie

**Przykład**     {{uwagi}}

: (1.1) forma ''tylni'' dla przymiotnika rodzaju męskiego w liczbie pojedynczej jest błędna, może odnosić się ona jedynie do liczby mnogiej  
<ref>{{PoradniaPWN|id=9687|hasło=tylny czy tylni?}}</ref>

\*

<b>Szablon</b>	{{tłumaczenia}}
<b>Zawartość</b>	Podsekcja ta pełni funkcję słownika z języka polskiego na inne. Podawane są odnośniki do wyrazów będących tłumaczeniami danego słowa polskiego.
<b>Języki</b>	tylko polski
<b>Przykład</b>	<div>{{tłumaczenia}}</div> <div><ul style="list-style-type: none"><li>* angielski: (1.1) [[date]], [[appointment]]</li><li>* arabski: (1.1) [[تَعْيِين]]</li><li>* francuski: (1.1) [[rendez-vous]]</li><li>* rosyjski: (1.1) [[свидание]], {{pot}} [[свиданка]]</li><li>* szwedzki: (1.1) [[träff]] {{w}}</li></ul></div>

\*

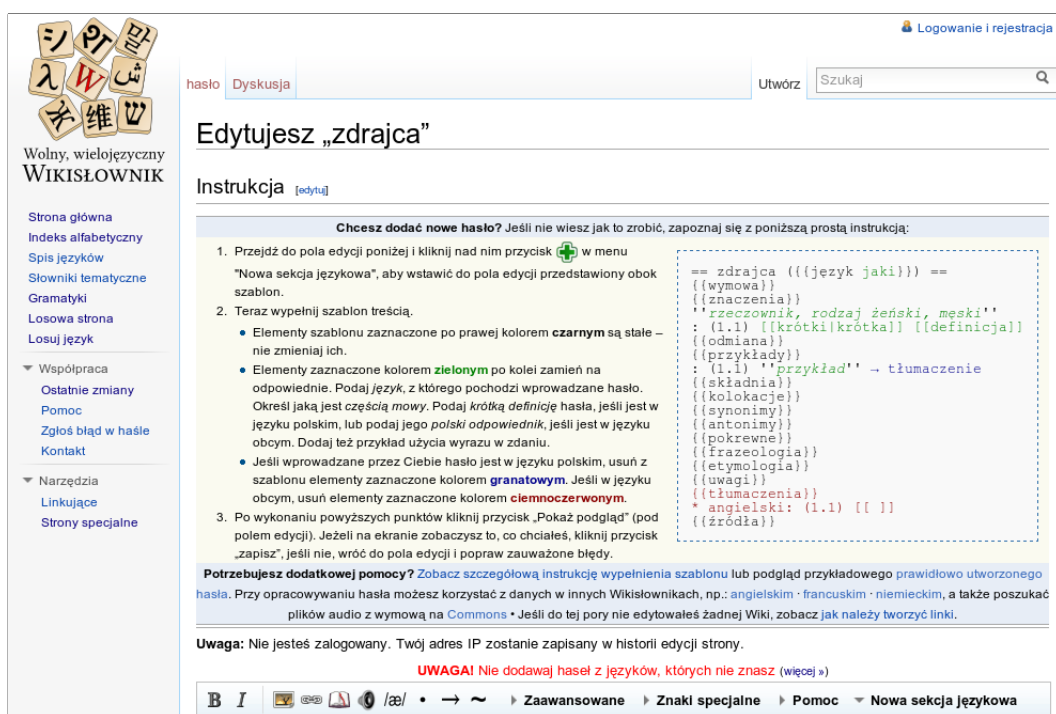
<b>Szablon</b>	{{źródła}}
<b>Zawartość</b>	Źródła dla informacji podanych w haśle. Zazwyczaj sekcja ta składa się ze znacznika <references/>, który powoduje wyświetlenie w tym miejscu przypisów wstawionych w poprzedzającej zawartości strony znaczników <ref>...</ref>.
<b>Języki</b>	wszystkie
<b>Przykład</b>	<div>{{źródła}}</div> <div>&lt;references/&gt;</div>

## 2.4.2. Wady dotychczasowych rozwiązań

W dotychczasowym systemie edycji w Wikisłowniku przedstawiona wyżej regularna struktura nie jest wykorzystywana w wystarczającym stopniu. W dużej mierze wiąże się to z pierwotnym założeniem, jakim jest oparcie projektu na silniku MediaWiki. Każde hasło przechowywane jest w bazie danych jako jeden ciąg znaków, a ich struktura jest wymuszona dopiero przez społeczność — brakuje jakichkolwiek mechanizmów, które pomagałyby w utrzymywaniu haseł zgodnie ze standardem.

Baza Wikisłownika nie zachowuje nawet pierwszej postaci normalnej [15] — hasła nie są przecież danymi atomowymi. To duża wada projektu takiego jak słownik internetowy,

w którym dane powinny być prezentowane możliwie jednolicie. Postaci normalne pozwalają też na łatwiejszą modyfikację i obróbkę danych. Niestety niemożliwa jest zmiana silnika MediaWiki, dlatego należy dbać o standaryzację artykułów za pomocą innych środków. Aby pomieścić i logicznie powiązać informacje w każdym haśle, konieczne jest użycie omówionej struktury — sekcji i podsekcji, z zastosowaniem licznych szablonów. Edycja takiego hasła (p. ilustracja 2.4 na str. 16) będzie bardzo trudna dla początkującego redaktora Wikisłownika. Jeszcze większe problemy występują przy tworzeniu nowych haseł. Ilustracja 2.5 pokazuje ekran służący do tego celu (wywoływany np. po kliknięciu w czerwony link w dowolnym haśle, oznaczający nieistniejący do tej pory artykuł).



Ilustracja 2.5: Próba stworzenia nowego hasła w polskim Wikisłowniku

Jak widać na rzucie ekranu, sama instrukcja tworzenia nowego hasła jest na tyle skomplikowana, że może zniechęcić do edycji nowego redaktora. Wymagane jest skopiowanie podanego kodu i pieczołowite podmienianie poszczególnych jego elementów. Niestety tylko w bardzo ograniczonym stopniu uwzględniane jest, że niektóre języki mają zdecydowanie inną strukturę niż przykładowa. Osoba chcąc dodać do Wikisłownika nowy znak chiński będzie musiała nie tylko zapoznać się z regułami dla tej specyficznej odmiany haseł, ale też prawdopodobnie skopiować kod istniejącego hasła z tej kategorii. O wiele lepsza byłaby sytuacja, w której po wyborze języka edytujący otrzymuje odpowiedni szkielet hasła, który

będzie mógł uzupełnić standardowym formularzem, bez potrzeby wybierania fragmentów, które należy zmienić lub usunąć.

Oparcie projektów Wikimedia na systemie MediaWiki i wikikodzie powoduje tego typu problemy oczywiście nie tylko w Wikisłowniku. Także w Wikipedii komplikacje techniczne są dużą barierą dla nowych użytkowników, najczęściej przyzwyczajonych do narzędzi typu WYSIWYG (*What You See Is What You Get*) takich jak Microsoft Word. W badaniach przeprowadzonych w 2009 roku czytelnikom wszystkich wersji językowych Wikipedii zadano pytanie o powody nieuczestniczenia w pracach edycyjnych. 25,18% spośród 21 492 ankietowanych odpowiedziało *nie wiem, jak*, 12,49% stwierdziło, że zbyt słabo zna używane technologie, natomiast 24,43% wskazało, że obawiałoby się popełnienia błędu i „wpadnięcia w kłopoty” z tego powodu (najczęściej wymienianym powodem było *myślę, że nie mam wystarczającej wiedzy, którą mogę się podzielić* — 51,98%) [16]. Wynika z tego, że aż jedna czwarta czytelników, którzy wypełnili ankietę, to potencjalni redaktorzy, którzy nie angażują się w tworzenie encyklopedii ze względu na uwarunkowania techniczne. Brak podobnych badań przeprowadzanych na użytkownikach Wikisłownika, można jednak przypuszczać, że kształtują się one podobnie. Dodatkowym czynnikiem jest specjalizacja projektu — w pracach przy słowniku w dużej części uczestniczą lingwiści, którzy nie muszą biegle posługiwać się platformą MediaWiki.

W przypadku Wikipedii od dawna podejmowane są wysiłki w kierunku poprawienia obsługi technicznej serwisu. Istnieje np. alternatywny edytor haseł *wikEd* dostępny dla każdego użytkownika po wybraniu w preferencjach, w ograniczonym zakresie wspomagający redagowanie haseł [17]. Prawdopodobnie jednak edytor WYSIWYG wspólny dla przedsięwzięć Fundacji Wikimedia nie powstanie w przewidywalnej przyszłości. Z tego powodu korzystne jest usprawnianie poszczególnych projektów z uwzględnieniem ich cech charakterystycznych. Jednym z celów opisanej aplikacji jest zwiększenie komfortu korzystania z formularza wprowadzania haseł w Wikisłowniku. Implementacja tego projektu została omówiona w sekcji 4.2.

Drugą znaczącą wadą Wikisłownika jest wynikająca z wielojęzyczności ogromna redundancja, powodująca spore problemy dla czytelnika. Każda wersja językowa stanowi odrębną całość. Choć słowników internetowych nierzadko używają poliglotti, którym może być obojętne, w jakim języku zostaną zaprezentowane informacje, struktura Wikisłownika nie pozwala im wykorzystać swoich umiejętności. Specjalistyczne hasła mogą być dostępne

np. wyłącznie w jednej wersji językowej — przeszukiwanie poszczególnych edycji to marnotrawienie czasu, dlatego często najlepszym rozwiązaniem bywa po prostu użycie wyszukiwarki, np. Google. Informacje w różnych językach pojawiają się niezależnie od siebie, bowiem znakomita większość redaktorów jest aktywna wyłącznie w jednym projekcie. Przy tym zdarza się, że wprowadzający informacje wykonuje zupełnie niepotrzebną pracę, którą ktoś wykonał wcześniej w innej wersji językowej, a nawet w tej samej wersji, lecz podczas tworzenia innego hasła, w pewien sposób powiązanego z obecnie edytowanym.

Ważną funkcję we wszystkich projektach Wikimedia pełnią boty, odpowiadające w największych projektach za od 10 do 30% wszystkich edycji haseł [18]. W polskim Wikisłowniku programy te przede wszystkim uzupełniają linki interwiki, ale także wykonują bardziej skomplikowane akcje: standaryzują układ podsekcji i format linków, aktualizują indeksy i inne listy słów oraz dodają polską wymowę [19] [13]. Część z tych edycji jest wtórna i spowodowana błędami popełnianymi przez początkujących redaktorów — można by ich zatem uniknąć, usprawniając formularz edycji.

Automatyzacja edytora pozwala na sprawniejszą standaryzację i użycie informacji wprowadzonych już do innych wersji językowych Wikisłownika. Bardzo pomocne okazuje się tu API MediaWiki [20]. Ten interfejs programistyczny pozwala na sprawne pobieranie treści wybranych haseł z dowolnego projektu Wikimedia w celu dalszego przetwarzania. Zostało to wykorzystane podczas implementacji drugiej istotnej części aplikacji, a następnie opisane w sekcji 4.3.

### **2.4.3. Różnice w stosunku do Wikipedii**

Po analizie uwidacznia się odmienny charakter Wikisłownika w stosunku do Wikipedii, która była pierwszym projektem Fundacji i dziś jest jej sztandarowym przedsięwzięciem. W tabeli 2.2 zostały ujęte najważniejsze różnice między witrynami w aspekcie technicznym.

To wszystko powoduje, że usprawnienie edycji w Wikisłowniku jest o wiele łatwiejsze niż w Wikipedii, której bardzo złożona struktura w praktyce uniemożliwia dziś kompleksową przebudowę procesu redakcji. Ściśle określona struktura to ogromna zaleta projektu. Co ciekawe — nie jest ona charakterystyczna dla wszystkich jego wersji językowych. Po części prac włożonych w implementację okazało się, że polski Wikisłownik jest pod tym względem wyjątkowo dobrze zorganizowany. Wprawdzie wszystkie większe edycje zachowują

<b>Wikipedia</b>	<b>Wikisłownik</b>
Struktura hasła różnorodna	Bardzo ściśle określona struktura hasła
Przede wszystkim ciągły tekst	Listy, wypunktowania
Umiarkowana liczba szablonów używanych w hasłach w stosunku do pozostałej treści	Bardzo duży udział szablonów w treści hasła
Duże różnice pomiędzy wersjami językowymi	Małe różnice pomiędzy wersjami językowymi
Linki interwiki łączą informacje o tym samym obiekcie w różnych językach	Linki interwiki łączą informacje o tym samym słowie występującym w wielu językach, objaśnione w różnych językach
Nagłówki dzielą artykuł na sekcje o dowolnej semantyce	Nagłówki dzielą artykuł na sekcje odpowiadające poszczególnym językom

Tabela 2.2: Porównanie Wikipedii i Wikisłownika — aspekty techniczne. Zob. też tabelę 3.1

standardowy podział na sekcje językowe, jednak owe sekcje niekiedy przybierają dość oryginalne postaci. Po dokładniejszych badaniach można stwierdzić, że polski Wikisłownik jest wyjątkowo jednolity, co ma niebagatelne znaczenie zarówno dla czytelnika, jak i dla edytora. W kolejnym rozdziale zanalizowano m.in. specyfikę Wikisłownika, jeśli skupić się na aspektach społecznościowych. Jak się okaże, także te elementy powinny mieć pozytywny wpływ na możliwość wprowadzenia usprawnień.



## Rozdział 3

# Aspekty społecznościowe

Tematem tego rozdziału są społecznościowe aspekty zarówno samego Wikisłownika, jak i niniejszej pracy magisterskiej. Na początku omówiono podstawowe założenia koncepcji *wiki*, na której opiera się Wikisłownik. Kolejna część rozdziału to krótka analiza społeczności rozwijającej projekt. Na dalszych stronach znalazły się treści ściśle związane z tworzoną aplikacją: opis rozwoju oprogramowania dla specyficznego klienta, jakim jest szersza społeczność, oraz szczegółowa analiza jego wymagań.

### 3.1. Koncepcja *wiki*

Pojęcie *wiki* określa witrynę internetową umożliwiającą tworzenie i edycję dowolnej liczby połączonych ze sobą hiperłączami stron przez przeglądarkę internetową, za pomocą uproszczonego języka znaczników lub edytora WYSIWYG [21]. Opisanie w rozdziale 2 projekty Fundacji Wikimedia, szczególnie Wikipedia, są z pewnością najbardziej znanymi i największymi przykładami *wiki*.

Wiki to nie tylko technologia, ale także, a może przede wszystkim, społeczność. Rozmiary *wiki* rozciągają się od małych, zamkniętych witryn firmowych do największej Wikipedii. To, co łączy wszystkie tego typu projekty, to otwarty dostęp do edycji i tworzenia stron w obrębie danej grupy. W przypadku projektów Fundacji grupę tę stanowi niemal cała ludność świata dysponująca dostępem do internetu — poza nielicznymi krajami, w których sieć jest cenzurowana. Wśród redaktorów Wikipedii czy Wikisłownika nie istnieje żadna skodyfikowana hierarchia, co oznacza, że niezalogowany nowicjusz może zmienić stronę

dokładnie tak samo jak administrator projektu. Założenie to od lat stanowi główny zarzut pod adresem Wikipedii — negatywne opinie o tej encyklopedii są szeroko rozpowszechnione [22]. Mimo tego projekty stale rozwijają się i podnoszą swoją jakość — już w 2005 roku w czasopiśmie „Nature” opublikowano kontrowersyjne porównanie Wikipedii z *Encyclopædia Britannica*, z którego wynikało, że obie encyklopedie stoją na podobnym poziomie [23]. Jasno widać, że znaczenie projektów Fundacji Wikimedia jest dziś bardzo duże.

W tworzeniu Wikipedii i Wikisłownika biorą udział przedstawiciele całego społeczeństwa — w pracach uczestniczą zarówno profesorowie, jak i studenci. Jak wspomniano w sekcji 2.4.2, dla wielu z nich barierą jest poziom skomplikowania oprogramowania MediaWiki. Podczas gdy wielu uczestników, najczęściej tych o zainteresowaniach ścisłych, nie ma problemów z opanowaniem techniki edycji haseł, istnieje duża grupa osób, które deklarują, że mogłyby zaangażować się bardziej, gdyby nie konieczność opanowania złożonych sposobów wprowadzania informacji. Dzięki regularnej strukturze akurat polski Wikisłownik jest tym projektem, w którym uproszczenie tworzenia i edycji haseł jest najbardziej możliwe do przeprowadzenia. Ponieważ proces taki wymaga wspólnej decyzji ogółu społeczności, należało zanalizować, jak kształtują się interakcje między użytkownikami w projekcie.

### 3.2. Społeczność polskiej edycji Wikisłownika

W podsekcji 2.4.3 przedstawione zostały różnice między technicznymi elementami polskich wersji Wikipedii i Wikisłownika. Wysokie poziom uporządkowania haseł w Wikisłowniku daje nadzieję na wprowadzenie ulepszeń — główną potencjalną barierą może być więc społeczność, która stara się podejmować decyzje konsensualnie.

Dobrym sposobem przedstawienia społeczności Wikisłownika jest przeciwstawienie jej grupy osób edytujących w Wikipedii<sup>1</sup>. W tabeli 3.1 przedstawiono najbardziej widoczne różnice między nimi, które można zauważyć już po kilkugodzinnej lekturze archiwalnych dyskusji i historii edycji.

Można uznać, że w przypadku polskich wersji językowych Wikisłownik jest projektem znacznie spokojniejszym niż Wikipedia, której powszechność powoduje nieuniknione problemy, takie jak zaangażowanie osób chcących wykorzystać projekt do celów marketingowych lub ideologicznych czy liczne wandalizmy w wykonaniu znudzonych nastolatków.

---

<sup>1</sup>Od tej pory pod pojęciami *Wikipedia* i *Wikisłownik* domyślnie będą kryć się wersje polskojęzyczne.

<b>Wikipedia</b>	<b>Wikisłownik</b>
Dużo wandalizmów (edycji wykonywanych w złej wierze)	Bardzo mało wandalizmów
Kilkuset stale aktywnych redaktorów	Kilkunastu stale aktywnych redaktorów
Wielu użytkowników mających kłopoty z dostosowaniem się do zasad	Niewielu użytkowników mających kłopoty z dostosowaniem się do zasad
Liczne wojny edycyjne (wzajemne cofanie swoich edycji przez co najmniej dwóch redaktorów)	Praktycznie brak wojen edycyjnych
Niezliczone strony archiwalnych dyskusji	Łatwy dostęp do archiwalnych dyskusji
Dobra dokumentacja i pomoc	Dobra pomoc, ale szczegółowa dokumentacja bardzo uboga
Bardzo zróżnicowane obszary zainteresowania	Społeczność o wspólnych zainteresowaniach lingwistycznych

Tabela 3.1: Porównanie Wikipedii i Wikisłownika — aspekty społecznościowe. Zob. też tabelę 2.2

Społeczność redaktorów słownika zdecydowanie skupiona jest na stałym udoskonalaniu projektu. Z racji jego rozmiaru łatwiej jest o konsensus w dyskusjach, a dzięki mniejszej popularności innowacyjne rozwiązania techniczne mają większą szansę na realizację.

W polskim Wikisłowniku działa 26 administratorów, spośród których 14 jest aktywnych. Kilku z nich obsługuje swoje własne boty [8]. Dyskusje na ogólne tematy toczą się w tzw. *Barze* (odpowiedniku *Kawiarenki* w Wikipedii). Na podstawie strony specjalnej *Ostatnie zmiany* można oszacować poziom aktywności redaktorów w stosunku do Wikipedii. Podczas gdy tam ostatnie 500 zmian obejmują nieco ponad godzinę, w Wikisłowniku jest to więcej niż doba. Charakterystyczny jest dużo większy udział nowo tworzonych haseł w stosunku do poprawek w starszych artykułach niż w Wikipedii. To dość naturalne — hasła słownikowe są prostsze i często nie wymagają dalszych poprawek technicznych po ich utworzeniu. Dużą część haseł tworzy mała, najbardziej aktywna liczba użytkowników: dłuższa obserwacja pozwala na stwierdzenie, że proporcje są mniej więcej zgodne z zasadą Pareto (20% obiektów jest związanych z 80% zasobów). Wynika z tego, że warto stworzyć aplikację, która będzie służyć zarówno nowym użytkownikom (czego skutkiem będzie zwiększenie liczby zaangażowanych osób), jak i tym doświadczonym, bardziej aktywnym (większa wydajność i satysfakcja z użytkowania).

### 3.3. Analiza wymagań

Aby sprecyzować elementy, które złożą się na implementację aplikacji, konieczne jest przeprowadzenie analizy wymagań. Ten proces pozwoli na zrozumienie dokładnych potrzeb, które powinny zostać spełnione, i rozbicie ich na konkretne, jasno zdefiniowane wymagania [24]. Klasyczne metody powinny być jednak nieco zmodyfikowane ze względu na specyfikę projektu — odbiorcą nie jest klient biznesowy, ale społeczność skupiona wokół otwartego projektu (zob. sekcja 3.4).

W przypadku niniejszego projektu specyfikowanie wymagań prowadzone było w ramach dyskusji w *Barze* na łamach Wikisłownika oraz prywatnych rozmów z redaktorami. Dodatkowo swoje opinie przekazało kilka osób niezwiązanych z projektem, mogących spojrzeć na zagadnienie z punktu widzenia nowego użytkownika.

Najważniejszym produktem, będącym wynikiem analizy wymagań, jest specyfikacja wymagań biznesowych. Zgodnie z klasycznym podziałem zostały one podzielone na wymagania funkcjonalne i нефункционалне.

#### 3.3.1. Definicje

- **Użytkownik** — osoba dokonująca edycji w Wikisłowniku.
- **Użytkownik zalogowany** — osoba zalogowana w Wikisłowniku i dokonująca w tym projekcie edycji.
- **Użytkownik niezalogowany** — osoba niezalogowana w Wikisłowniku i dokonująca w tym projekcie edycji. Takie edycje zostaną zapisane wraz z adresem IP użytkownika, co umożliwi częściową identyfikację autora.
- **Przestrzeń nazw** — grupa stron w Wikisłowniku o wspólnym prefiksie zakończonym dwukropkiem. Przykładowe przestrzenie nazw to *Szablon:*, *Dyskusja:*. Szczególnym przypadkiem jest tzw. przestrzeń główna, skupiająca strony bez dodatkowego prefiksu — w niej znajdują się wszystkie hasła słownika.
- **Hasło** — strona w przestrzeni głównej Wikisłownika.
- **Stary formularz** — dotychczasowa metoda wprowadzania danych, opisana w sekcji 2.4.1.
- **Nowy formularz** — aplikacja będąca przedmiotem niniejszej pracy.

- **Sekcja** — odcinek pojedynczego hasła dotyczący użycia słowa w pojedynczym języku, wyróżniony nagłówkiem drugiego stopnia (==).
- **Podsekcja** — element sekcji hasła zgodny z ogólną strukturą haseł (p. sekcja 2.4.1).
- **Interwiki** — linki pomiędzy poszczególnymi wersjami językowymi Wikisłownika.

### 3.3.2. Wymagania funkcjonalne

#### **Edycja istniejących haseł i sekcji za pomocą nowego formularza**

Użytkownik musi mieć możliwość wprowadzenia zmian w obrębie całego hasła Wikisłownika za pomocą nowego formularza. Formularz powinien umożliwiać edycję dowolnego elementu w hasle — w szczególności dotyczy to sekcji wstępnej, zawierającej dane ogólne, niezwiązane z żadnym językiem. Edycja ma polegać na uzupełnieniu wartości dla poszczególnych kluczy w formularzu. Kluczami są tytuły kolejnych podsekcji, wartościami ich zawartość.

#### **Dodanie nowej sekcji do istniejącego hasła za pomocą nowego formularza**

Użytkownik musi mieć możliwość rozszerzenia istniejącego hasła o kolejną sekcję językową. Sekcja powinna zostać automatycznie uzupełniona zawartością charakterystyczną dla wybranego języka.

#### **Utworzenie nowego hasła za pomocą nowego formularza**

Użytkownik musi mieć możliwość stworzenia nowego hasła, składającego się z jednej lub wielu sekcji językowych. Każda sekcja powinna być utworzona w sposób analogiczny do dodawania nowej sekcji do istniejącego hasła.

#### **Edycja i utworzenie nowego hasła za pomocą starego formularza**

Użytkownik, który chce korzystać nadal ze starego formularza, musi mieć taką możliwość. Decyzja o wyborze starego lub nowego formularza musi być łatwa do podjęcia i możliwa do odwrócenia w każdej chwili, także w trakcie edycji lub tworzenia hasła.

#### **Wprowadzanie znaków specjalnych**

Użytkownik musi mieć możliwość prostego wprowadzania do formularza znaków specjalnych, w szczególności liter alfabetów używanych w językach opisywanych przez Wikisłownik.

### **Usunięcie sekcji językowej**

Użytkownik musi mieć możliwość usunięcia dowolnej sekcji językowej z hasła.

### **Edycja tytułu sekcji językowej**

Użytkownik musi mieć możliwość zmiany tytułu sekcji językowej. Jest to konieczne z powodu odrębnego traktowania haseł opisujących przysłowia, związki frazeologiczne i inne wyrażenia składające się z kilku wyrazów.

### **Automatyczne pobieranie danych z innych wersji językowych**

Użytkownik musi mieć możliwość wyszukania w innych wersjach językowych Wikisłownika danych, które mogą być cennym uzupełnieniem edytowanego hasła. W szczególności dotyczy to ilustracji, wymowy w międzynarodowym alfabecie fonetycznym (IPA), nagrań dźwiękowych.

### **Automatyczne uzupełnianie podsekcji**

Aplikacja powinna automatycznie uzupełnić te spośród podsekcji w danym haśle, które to umożliwiają. W szczególności dotyczy to linków interwiki w nowo dodawanej sekcji wstępnej, szablonów używanych do transliteracji, znacznika <references/> w podsekcji *źródła*.

## **3.3.3. Wymagania niefunkcjonalne**

### **Licencjonowanie**

Aplikacja musi zostać udostępniona na licencjach GNU FDL 1.2 i CC-BY-SA 3.0 — wymagają tego zasady Wikisłownika i wszystkich projektów Wikimedia. W związku z powyższym cały kod użyty w aplikacji musi być dostępny już wcześniej na odpowiednich licencjach bądź udostępniony na nich przez autora pracy. Grafiki użyte w interfejsie użytkownika muszą pochodzić z serwisu Wikimedia Commons, zbierającego pliki na wolnych licencjach.

### **Integracja z Wikisłownikiem**

Aplikacja musi być w pełni zintegrowana z istniejącym interfejsem Wikisłownika. Z tego powodu jedyną możliwą opcją jest aplikacja kliencka napisana w JavaScript, w ramach

nie wchodzi natomiast aplikacja desktopowa wymagająca dodatkowych czynności. Korzystanie z nowego formularza powinno być inicjowane w dokładnie ten sam sposób co korzystanie ze starego formularza.

### **Pomoc kontekstowa**

Nowy formularz musi zawierać pomoc kontekstową dla wszystkich elementów, których działanie nie jest oczywiste. Wyświetlanie pomocy nie może utrudniać edycji hasła.

### **Interfejs użytkownika**

Interfejs użytkownika musi być prosty, czytelny i intuicyjny. W szczególności bardzo istotne jest, aby dla nowego użytkownika nowy formularz stanowił duże ułatwienie i zmniejszał barierę wejścia do projektu.

### **Obsługiwane oprogramowanie**

Ponieważ Wikisłownik musi być dostępny do edycji dla jak największej liczby użytkowników, jest bardzo ważne, aby aplikacja obsługiwała wszystkie popularne przeglądarki internetowe: Firefox  $\geq 4$ , Chrome 13 (dzięki automatycznej aktualizacji w przypadku tej przeglądarki nie występuje problem obsługi starszych wersji), Opera  $\geq 10$  i Internet Explorer  $\geq 7$ . W przypadku braku wyłączonej obsługi JavaScript zachowanie formularza ma zostać niezmienione: załaduje się stary formularz z wyłączonymi niektórymi funkcjami.

### **Wydajność**

Aplikacja musi umożliwiać płynną pracę — skrypty nie mogą być znacząco wolniejsze od skryptów użytych w starym formularzu.

### **Łatwość modyfikacji**

Ze względu na niestały charakter zasad panujących w Wikisłowniku kod aplikacji musi być dostępny publicznie i możliwy do modyfikacji w każdej chwili. W szczególności dotyczy to komunikatów dla użytkownika i używanych stałych, które powinny być zebrane w jednym miejscu.

### 3.4. Specyfika tworzenia aplikacji dla wikispółeczności

Opisywana aplikacja jest projektem dość szczególnym: nie powstaje na zamówienie klienta biznesowego, który mógłby określić dokładne wymagania. Ma także mało wspólnego z klasycznymi projektami *open source* — z założenia jej autorem jest jedna osoba, autor niniejszej pracy. Projekt formularza przyjmował coraz bardziej sprecyzowany kształt dzięki wypowiedziom członków otwartej społeczności i ich wzajemnej dyskusji, duże znaczenie miały też testy wykonywane przez ochotników z Wikisłownika. Aplikacja powstawała inkrementalnie, co kilka dni–tygodni skrypt na stronach Wikisłownika aktualizowany był do kolejnej działającej wersji z nowymi funkcjami.

Dzięki takiemu modelowi budowy aplikacji na bieżąco wykrywane były błędy w działaniu, o które było łatwo ze względu na poziom skomplikowania kluczowych modułów takich jak parser wikitekstu. Projekt okazał się ciekawym doświadczeniem łączącym pozytywne cechy tworzenia oprogramowania dla klienta biznesowego i programu na swój własny użytek.

Zarys projektu powstał jesienią 2010 roku w wyniku pierwszej dyskusji w *Barze Wikisłownika*. Założona została wówczas podstrona poświęcona dyskusji wyłącznie temu zagadnieniu — rozmowy w ramach *Baru* byłyby raczej niewygodne. Główny etap rozwoju aplikacji przypadł na okres od maja do września 2011 roku. W toku dyskusji kilkakrotnie zmieniały się wymagania klienta, jakim jest wikispółeczność. W tym wypadku, w odróżnieniu od projektów biznesowych, cecha ta okazała się pozytywna. Dzięki bieżącym konsultacjom końcowy kształt aplikacji był satysfakcjonujący dla wszystkich stron uczestniczących w przedsięwzięciu. Sekcja 4.4 poświęcona jest ostatnim etapom projektu: wdrożeniu w Wikisłownika i widokom na dalszy rozwój.



## Rozdział 4

# Opis implementacji

W ostatnim rozdziale szczegółowo przedstawiony została implementacja aplikacji. Opisano użyte technologie i metodologię, a szczególną uwagę poświęcono problemom, które powstały podczas procesu tworzenia programu. Pierwsza sekcja zawiera opis środowiska, w którym osadzona jest aplikacja, natomiast kolejne dwie skupiają się na dwóch logicznie wyodrębnionych częściach projektu: uproszczonym formularzu edycyjnym oraz automatyzacji tworzenia haseł. W ostatniej części rozdziału przedstawiono przebieg wdrożenia aplikacji w polskim Wikisłowniku i możliwości dalszego rozwoju.

Niniejszy rozdział zawiera odniesienia do plików źródłowych składających się na aplikację oraz pomocniczych. Wszystkie te pliki znajdują się na dołączonej do pracy płycie CD w katalogu src.

### 4.1. Wprowadzenie

Oprogramowanie MediaWiki zostało napisane w języku PHP (*PHP Hypertext Preprocessor*) i jest wysoce konfigurowalne za pomocą tzw. rozszerzeń, dodających kolejne funkcje — przykładowo: obsługę przypisów, dodatkowe strony specjalne czy nowe funkcje parsera [25]. W projektach Fundacji obsługa rozszerzeń jest kontrolowana przez Fundację. Poszczególne projekty mogą składać prośby o włączenie określonego rozszerzenia, decyzję podejmuje zaś główni programiści Fundacji. Teoretycznie można rozważać utworzenie aplikacji dla Wikisłownika jako rozszerzenia w PHP. Opcja ta została jednak odrzucona we wstępnej fazie projektu — stworzenie rozszerzenia wymagałoby nieporównanie więcej formalności

niż aplikacja kliencka, przede wszystkim jednak jego funkcjonalność byłaby znacznie ograniczona: poza polskim Wikisłownikiem żaden inny projekt nie mógłby z niego skorzystać, na pewno nie zostałby więc włączony do głównej wersji MediaWiki.

#### 4.1.1. JavaScript w Wikisłowniku

Podobnie jak w wielu innych przypadkach w obrębie projektów Wikimedia jako metodę dostosowania silnika MediaWiki do szczególnych potrzeb wybrano zatem aplikację napisaną w języku JavaScript. Konieczne jest zatem przedstawienie sposobu obsługi skryptów na stronach tych witryn. Pliki JavaScript ładowane są z wielu różnych źródeł — hierarchia w nieco uproszczonej postaci jest następująca [26]:

1. skrypty systemowe oprogramowania MediaWiki, wspólne dla wszystkich projektów oraz charakterystyczne dla danego projektu ze względu na ładowane rozszerzenia,
2. skrypt zapisany na poziomie danego projektu jako strona `MediaWiki:Common.js` — dostęp do niego mają lokalni administratorzy,
3. skrypt zapisany na poziomie danego projektu dla wybranej przez użytkownika skórki (domyślną jest *Vector*, poprzednią był *Monobook*) jako strona, np. `MediaWiki:Vector.js` — dostęp do niego mają lokalni administratorzy,
4. tzw. gadżety, czyli dodatkowe skrypty rozszerzające funkcjonalność, możliwe do włączenia w preferencjach użytkownika — także one edytowane i wybierane są przez administratorów,
5. skrypt zapisany na stronie danego użytkownika, np. `User:Sokrates/common.js` — na tym poziomie możliwe jest dostosowywanie skryptów do swoich potrzeb przez każdego użytkownika,
6. skrypt zapisany na stronie danego użytkownika dla wybranej skórki, np. dla skórki *Vector* — `User:Sokrates/vector.js`.

Jak widać, możliwości dołączania dodatkowych skryptów są elastyczne i dostępne na różnych poziomach dla różnych użytkowników. Decyzja o włączeniu skryptów dla wszystkich użytkowników danego projektu należy do jego administratorów (nie licząc wspólnych skryptów narzuconych przez Fundację), natomiast każdy użytkownik poprzez edycję specjalnej strony może uruchamiać różne skrypty na swoje potrzeby, nie będąc zmuszonym

do korzystania z dodatków do przeglądarek, takich jak Greasemonkey. W ten sposób ułatwiony jest proces powstawania i testowania nowych skryptów: osoby chcące przetestować nowy skrypt mogą na swojej stronie JS dołączyć go do swojej. Jeśli administratorzy uznają skrypt za wartościowy, mogą go dodać do zbioru gadżetów (wtedy każdy użytkownik ma możliwość włączenia skryptu bez konieczności edycji plików JS) lub do ogólnego pliku `MediaWiki:Common.js`.

## **MediaWiki 1.17**

Projekty Fundacji od czerwca 2011 roku używają wersji MediaWiki 1.17. Wersja ta wprowadziła kilka istotnych zmian, spośród których najważniejszą było wprowadzenie modułu `ResourceLoader`, zmieniającego system ładowania dodatkowych plików, przede wszystkim skryptów i arkuszy stylów [27]. Wszystkie potrzebne pliki są łączone w jeden i dodatkowo kompresowane, dodatkowo poprawiono też obsługę pamięci podręcznej po stronie klienta. Do tej pory poszczególne pliki JS i CSS przesyłane były statycznie w wielu żądaniach HTTP, co mogło powodować problemy z ich synchronizacją, a przede wszystkim zwiększało obciążenie sieci. Aktualizacja software'u projektów Fundacji spowodowała konieczność aktualizacji większości używanych w nich skryptów i zaangażowała większość aktywnych uczestników zaznajomionych z technikami [28].

Od wersji 1.16 do MediaWiki dołączana jest biblioteka programistyczna jQuery, a w najnowszym wydaniu zaktualizowano ją do wersji 1.4.2 [29]. Jest to jedna z najczęściej używanych bibliotek dla JavaScript, w bardzo dużym stopniu usprawniająca programowanie w tym języku. Kod pisany za jej pomocą jest o wiele czytelniejszy i prostszy, biblioteka pozwala też wyeliminować wiele problemów związanych z nieprawidłową obsługą JavaScript w niektórych przeglądarkach [30] [31].

### **4.1.2. Środowisko programistyczne**

Programowanie w JavaScript wymaga użycia innych metod niż tworzenie aplikacji desktopowych. Do tworzenia edytora dla Wikisłownika użyto m.in. następujących aplikacji:

#### **Geany**

Lekkie zintegrowane środowisko programistyczne (IDE) [32]. W początkowej fazie projektu używane było środowisko Eclipse z dodatkowymi wtyczkami — okazało się jednak, że

obsługa JavaScript nie jest w nim na tyle wygodna, aby konieczne było wykorzystywanie tak rozbudowanego edytora.

### **Przeglądarki internetowe**

Aplikacja została przetestowana w przeglądarkach Firefox  $\geq 4$ , Chrome 13, Opera  $\geq 10$  i Internet Explorer  $\geq 7$ , pod systemami operacyjnymi Linux i Windows. Wyeliminowano błędy, które zostały odkryte w czasie testów. W trakcie implementacji używane były przeglądarki Google Chrome (i jej wbudowane narzędzia deweloperskie) oraz Firefox z dodatkiem Firebug, ułatwiającym debugowanie aplikacji [33]. Prawidłowe działanie aplikacji pod maksymalną możliwą liczbą dostępnych przeglądarek jest kluczowe ze względu na jej charakter — edycja jest dostępna dla każdego użytkownika. Nie testowano jednak edytora w starszych wersjach tych programów ze względu na ich znikome rozpowszechnienie i niewspółmierny koszt takich testów w stosunku do spodziewanych korzyści.

### **Git**

Systemem kontroli wersji używanym w projekcie był Git. Dzięki niemu możliwe było uniknięcie jakichkolwiek problemów związanych z zarządzaniem wersjami. Dla bezpieczeństwa kod był na bieżąco wysyłany do zdalnego repozytorium założonym w serwisie GitHub [34]. Kontroli wersji podlegała również niniejsza praca i jej kod źródłowy w systemie  $\text{\LaTeX}$ . Historia zmian aplikacji i pracy jest dostępna do wglądu na serwerach GitHub [35].

### **The Regex Coach**

W aplikacji duże znaczenie odgrywają wyrażenia regularne, wykorzystywane podczas parsowania wikitekstu. The Regex Coach jest programem ułatwiającym konstruowanie poprawnych wyrażeń [36].

\*

#### **4.1.3. Dobre praktyki**

Nowy edytor został napisany za pomocą biblioteki jQuery, co ułatwia utrzymanie dobrej struktury. Kod podzielono na moduły funkcjonalne, co częściowo stanowi odpowiednik klasycznego paradygmatu obiektowego, często stosowany w przypadku JavaScript. Język ten umożliwia pisanie aplikacji przy zachowaniu wielu różnych paradygmatów. W przypadku opisywanej aplikacji zdecydowano się na wspomniany podział na moduły, będący

pewnym kompromisem między paradygmatem funkcyjnym i obiektowym. Moduły zaimplementowane są jako obiekty, których elementami są funkcje odpowiedzialne za operacje działające w podobnym zakresie. Przykładowo parser wikitekstu jest modulem o następującej strukturze:

```
1 var EParser = {
2   getSections : function (code) {
3     ...
4   },
5   getSectionFromTitle : function (str) {
6     ...
7   },
8   getTitleFromCode : function (code) {
9     ...
10  },
11  ...
12 };
```

Listing 4.1: Ogólna struktura modułu JavaScript

Podział na moduły ułatwia uwzględnienie niektórych spośród licznych praktyk zalecanych przy programowaniu w JavaScript. Istnieje kilka narzędzi, które w automatyczny sposób badają jakość kodu w tym języku, pomagając znaleźć potencjalne błędy i uniknąć kolejnych. Podczas implementacji stale korzystano z trzech z nich: JSHint [37], JavaScriptLint [38], a przede wszystkim JSLint [39]. Narzędzia te nie są do końca kompatybilne ze sobą nawzajem, dlatego za cel obrano całkowity brak ostrzeżeń generowanych przez JSLint. W ten sposób udało się utrzymać m.in. następujące elementy:

- kontrola nad zmiennymi globalnymi (w przeciwieństwie do innych języków, np. PHP, domyślna deklaracja zmiennej bez słowa kluczowego `var` powoduje umieszczenie jej w zakresie globalnym),
- kończenie poleceń średnikiem (ich brak może prowadzić do błędów bardzo trudnych do wykrycia),
- odpowiednie tworzenie bloków kodu,
- sprawdzanie `if (object.hasOwnProperty(name))` wewnątrz pętli `for name in object` (z powodu rozszerzalności prototypów funkcji brak takiego obostrzenia prowadzi do błędów),
- prawidłowe używanie operatorów takich jak `===`, `!==`,

- prawidłowe używanie konstruktorów,
- spójne wcięcia i używanie nawiasów.

Istotnym zagadnieniem jest podział kodu na pliki. Podczas gdy w trakcie implementacji oczywiste jest, że podział na moduły powinien implikować także podział na pliki, na cele testowania aplikacji przez społeczność o wiele wygodniejsze jest używanie jednego pliku zawierającego jej całość — w przeciwnym razie konieczne byłoby stałe edytowanie większej liczby stron specjalnych Wikisłownika. Używanie wielu plików powoduje też problemy z synchronizacją ich ładowania, ponieważ funkcja inicjalizująca aplikację powinna zostać uruchomiona dopiero po wczytaniu całości aplikacji.

Po rozważeniu kilku możliwości zdecydowano się na użycie skryptu `make.sh` napisanego w Bashu, łączącego wszystkie pliki aplikacji w jeden, który okresowo był aktualizowany na stronie <http://pl.wiktionary.org/wiki/Wikipedysta:ToSter/ed.onefile.js>. Dodatkowym problemem było dołączenie kodu CSS: standardową metodą używaną w Wikisłowniku byłoby utworzenie oddzielnej strony zakończonej na `.css` i ładowanie jej razem ze stroną `.js`. W skrypcie budującym wynikowy kod użyto jednak innej techniki — arkusz stylów z lokalnego pliku `.css` był przetwarzany w sposób umożliwiający dynamiczne dodanie go do zawartości strony jako łańcuch znaków. Skrypt `make.sh` wykonywał też kilka innych czynności: opatrywał program adnotacjami dla JSLint, usuwał zbędne spacje i znaki BOM w plikach Unicode oraz zamykał całość aplikacji w domknięciu JavaScript (*closure*).

Kod aplikacji jest w maksymalnie możliwym zakresie zgodny z konwencjami przyjętymi przez programistów MediaWiki na potrzeby wewnętrznych części systemu [40].

## 4.2. Formularz edycyjny

Jedną z dwóch głównych części aplikacji jest nowy formularz, usprawniający w znacznym zakresie proces edycji i tworzenia nowych artykułów. Do tej pory cały ów proces polegał na wpisywaniu, kopiowaniu i uzupełnianiu wikikodu (por. ilustracja 2.4), a najprostszym rozwiązaniem często okazywało się kopiowanie go z innego, podobnego hasła i modyfikowanie jedynie zmieniających się elementów. Dlatego też większość haseł w Wikisłowniku powstawała seriami — tworzenie pojedynczych haseł było niepraktyczne, natomiast wstawianie podobnego kodu do dziesiątek haseł dość łatwe do przeprowadzenia. Jako że

każda sekcja językowa hasła składa się w rzeczywistości z par klucz–wartość, wygodniejszym sposobem wprowadzania danych byłby klasyczny formularz, do wypełniania którego przyzwyczajony jest każdy internauta. Nowy edytor został zatem zaprojektowany właśnie w taki sposób, dzięki czemu zmniejsza się bariera wejścia dla początkującego redaktora. Od tej pory edycja hasła będzie polegać nie na wpisywaniu dość skomplikowanego kodu, co jest porównywalne z prostym programowaniem, ale na wypełnieniu formularza.

Implementacja formularza składa się z kilku modułów, spośród których główną rolę odgrywają EUi w pliku `ed.ui.js` (interfejs użytkownika), EParser i ESectionParser w pliku `ed.parser.js` (parser wikitekstu) oraz EPrinter w pliku `ed.printer.js` (drukowanie danych przetworzonych przez aplikację). Poniżej opisano te moduły, szczególnie uwzględniając powstałe problemy i ich rozwiązania.

#### 4.2.1. Interfejs użytkownika

Kluczowym elementem projektu i implementacji aplikacji jest interfejs użytkownika — to jego jakość stanowi o końcowym powodzeniu przedsięwzięcia. Szczególnie ważna jest bezproblemowa integracja nowego formularza z dotychczasowym interfejsem. Wszystkie dotychczasowe funkcje używane w trybie edycji, których zastosowanie miałoby sens i w nowej wersji, muszą pozostać dostępne.

Zrzuty ekranu nowego formularza załączono w dodatku B.

Ilustracja 4.1: Kluczowe elementy interfejsu w trybie edycji

Podstawowym problemem, powstałym na początku implementacji, było podpięcie nowego formularza pod mechanizmy przetwarzające dane po stronie serwera. Z powodu braku jakiegokolwiek dostępu do konfiguracji serwera konieczne jest zapewnienie prawidłowego jego działania jedynie przy użyciu JavaScript. Na ilustracji 4.1 wyróżniono m.in. najważniejsze elementy sterujące edycją hasła. Są to trzy przyciski, których wybór powoduje wysłanie formularza w trybie POST na serwer do strony `index.php` — dalsze akcje uzależnione są od tego, który przycisk wybrano. Nową treścią hasła staje się zawartość głównego pola typu `<textarea>`, o identyfikatorze `name="wpTextbox1"`. W celu zapewnienia działania nowego formularza zgodnie z oprogramowaniem MediaWiki możliwe były dwie opcje:

1. Usunięcie standardowych elementów, przede wszystkim pola `wpTextbox1`, i zastąpienie ich własnymi. Konieczne byłoby stworzenie pola formularza o nazwie `wpTextbox1` — mogłoby to być pole ukryte (`<input type="hidden" />`), które musiałyby być synchronizowane z polami nowego formularza.
2. Ukrycie standardowych elementów i synchronizacja zawartości pola `wpTextbox1` z polami nowego formularza.

Wybór padł na rozwiązanie drugie z kilku powodów. Okazało się, że dostęp do standardowych elementów w trakcie działania nowego formularza jest bardzo przydatny, przede wszystkim zaś ich ukrywanie (za pomocą metody jQuery `.hide()`, zmieniającej atrybut CSS `display` na `none`) umożliwia proste przełączanie między starym a nowym formularzem w trakcie edycji, co było jednym z wymagań funkcjonalnych aplikacji. Należało zatem zapewnić prawidłową synchronizację pomiędzy polem `wpTextbox1` a polami nowego formularza. Najważniejsze było podpięcie odpowiednich zdarzeń pod przyciski wysyłające formularz na serwer, co widoczne jest na listingu 4.2.

```
1  rebindFormActions : function () {
2      this.form.find('textarea').removeAttr('name');
3      $('form').submit(function () {
4          if (EUi.usingNew) {
5              EUi.deleteEmptySections();
6              EUi.tbox.val(EPrinter.recalculateCode());
7          }
8          return true;
9      });
10 }
```

Listing 4.2: Funkcja `EUi.rebindFormActions`



W momencie próby wysłania formularza na serwer skrypt sprawdza, czy użyty jest nowy formularz. W takim przypadku ukryte okno tekstowe starego formularza, zapisane jako zmienna `EUi.tbox`, wypełniane jest nową zawartością. Za jej obliczenie odpowiada funkcja `EPrinter.recalculateCode`, opisana w podsekcji 4.2.2. Dzięki usunięciu atrybutów `name` ze wszystkich pól tekstowych w obrębie nowego formularza redukowana jest ilość przesyłanych danych, a z punktu widzenia serwera nic się nie zmienia.

Z kolei po zmianach wprowadzonych za pomocą starego formularza konieczne jest ponowne parsowanie wikikodu, które jest wykonywane w identyczny sposób jak przy inicjowaniu nowego edytora.

Ilustracja 4.2: Fragment nowego formularza w haśle z jedną sekcją językową

Zrzut ekranu 4.2 przedstawia podstawowe elementy interfejsu nowego formularza:

1. Link pozwalający na przełączanie pomiędzy nowym a starym edytorem,
2. Zakładki odpowiadające poszczególnym sekcjom językowym. Za podział na sekcje odpowiada moduł `EParser`, natomiast zadaniem modułu `EUi` jest ich odpowiednie wyświetlenie. Każdej sekcji odpowiada element `<fieldset>` zawierający pola tekstowe z poszczególnymi podsekcjami. Górne zakładki pozwalają na przełączanie pomiędzy sekcjami — w ten sposób redaktor widzi tylko tę sekcję, którą chce w danym momencie edytować. W starym formularzu konieczne było odnalezienie odpowiedniego kawałka kodu.

Sekcje językowe oznaczone są kodami ISO stosowanymi powszechnie w Wikisłowniku. Oczywiście nie wszystkie kody są intuicyjne, dlatego po najechaniu myszą na

każdą zakładkę pojawia się dokładna odpowiedź na temat danej sekcji.

Do zakładek językowych dodane są także zakładki o specjalnej funkcji: pierwsza z nich pozwala na edycję sekcji wstępnej (zawierającej m.in. interwiki), druga umożliwia dodanie nowej sekcji (pojawia się wówczas okno dialogowe z zapytaniem o język nowej sekcji).

3. Podręczna klawiaturka umożliwia wprowadzanie dużą liczbę znaków specjalnych do obecnie edytowanego pola. Podobna klawiaturka znajduje się w starym edytorze — została oznaczona kolorem żółtym na ilustracji 4.1. W nowym formularzu funkcja ta została usprawniona. Udało się wykorzystać istniejące już skrypty i zintegrować z aplikacją. Listing 4.3 przedstawia moduł ESpecialChars, odpowiadający za przemieszczanie elementów HTML pomiędzy nowym a starym formularzem.
4. Przyciski pod poszczególnymi polami formularza pozwalają na automatyzację pewnych akcji. Ich działanie opisano w sekcji 4.3.

```
1 ESpecialChars = {
2   /* element HTML zawierający klawiaturkę */
3   obj : undefined,
4   /* dotychczasowy element-rodzic klawiaturki */
5   formerParent : undefined,
6   detached : 0,
7
8   /* odłączenie klawiaturki od starego edytora i dodanie do elementu keyboard_keys w
       nowym formularzu */
9   detach : function () {
10      var container;
11      if (ESpecialChars.detached) {
12         return;
13      }
14      container = $('#keyboard_keys');
15      ESpecialChars.obj = $('#editpage-specialchars');
16      ESpecialChars.formerParent = ESpecialChars.obj.parent();
17      ESpecialChars.obj.detach();
18
19      container.append(ESpecialChars.obj);
20      ESpecialChars.detached = 1;
21   },
22
23   /* ponowne przyłączenie klawiaturki do starego rodzica */
24   attach : function () {
25      if (!ESpecialChars.detached) {
26         return;
27      }
28      EKeyboard.hide();
29      ESpecialChars.obj.detach();
30      ESpecialChars.formerParent.append(ESpecialChars.obj);
```

```
31     ESpecialChars.detached = 0;
32 },
33
34 toggle : function () {
35     if (ESpecialChars.detached) {
36         ESpecialChars.attach();
37     } else {
38         ESpecialChars.detach();
39     }
40 }
41 };
```

Listing 4.3: Moduł ESpecialChars

Na ilustracji 4.3 pokazano działanie klawiaturki ekranowej w nowym formularzu. Klawiaturę włącza się i wyłącza kliknięciem w ikonę z klawiaturą, widoczną pod aktywnym właśnie polem tekstowym. Oprócz zawartości dostępnej w starym formularzu można też wybrać jeden z najczęściej używanych znaków. Ta funkcja znacząco usprawniła proces tworzenia haseł w językach używających innych alfabetów lub znaków diakrytycznych. Stan klawiatury ekranowej (włączona/wyłączona) zapamiętywany jest w ciasteczku przeglądarki, zostaje więc zachowany na kolejnych stronach.

[illegible]

Ilustracja 4.3: Użycie klawiaturki ekranowej w nowym formularzu

Ciekawym problemem było ułożenie pól formularza w taki sposób, by zajmował możliwie mało miejsca, a przy tym pozostał przejrzysty. Każda sekcja językowa składa się z kilkunastu podsekcji, dlatego należało dobrze rozmieścić kilkanaście pól tekstowych, przy

czym te mogły zawierać bardzo dużo tekstu (np. podsekcja *tłumaczenia* w popularnych polskich hasłach) albo i pozostawać puste. Najprostszym, standardowym rozwiązaniem byłoby ustawienie stałej wysokości (np. równej 3 wierszom) dla każdego pola, które byłoby wyposażone w zwykły pasek przewijania. Takie pola w większości przypadków zajmowałyby jednak o wiele za dużo miejsca, przy tym formularz z dużymi białymi plamami jest niezbyt estetyczny. Zdecydowano się na rozwiązanie spotykane czasem w portalach internetowych, najlepiej znane z Facebooka — pola tekstowe zmieniają swój rozmiar w trakcie wpisywania kolejnych znaków. Istnieje kilka wtyczek do jQuery o podobnej funkcjonalności, jednak na potrzeby aplikacji konieczne było stworzenie takiej, która łączy cechy kilku z nich. Problemem okazała się też wydajność, ponieważ niektóre hasła mają w trybie edycji nawet ponad sto pól dynamicznie dostosowujących swój rozmiar. Rozwiązaniem jest dołączenie do strony ukrytego elementu `<div>`, do którego na bieżąco kopiowana jest zawartość ukrytego pola tekstowego — na podstawie jego wysokości ustawiana jest wysokość pola. Kod przedstawiony jest na listingu 4.4.

```
1  /* wzorowane na https://github.com/jaz303/jquery-grab-bag/raw/master/javascripts/
   jquery.autogrow-textarea.js */
2  (function ($) {
3      $.fn.autoresize = function () {
4          var shadow = $('<div/>').css({
5              position: 'absolute',
6              top: -10000,
7              left: -10000,
8              resize: 'none'
9          }).appendTo(document.body);
10
11      this.filter('textarea').each(function () {
12          var $this = $(this), minHeight = 25, maxHeight = 500,
13              prevHeight = 0, nowHeight = 0;
14          var update = function () {
15              var val = this.value.replace(/[\<>]/g, 'w').replace(/\n$/, '<br/>&nbsp;');
16              ;').replace(/\n/g, '<br/>');
17              shadow.html(val);
18              nowHeight = Math.min(Math.max(shadow.height(), minHeight), maxHeight);
19              if (nowHeight !== prevHeight) {
20                  $(this).css('height', nowHeight);
21                  EKeyboard.updatePosition($(this));
22                  prevHeight = nowHeight;
23              }
24          };
25          shadow.css({
26              width: $(this).width() - parseInt($this.css('paddingLeft'), 10) -
27                  parseInt($this.css('paddingRight'), 10),
28              fontSize: $this.css('fontSize'),
```

```

28         fontFamily: $this.css('fontFamily'),
29         lineHeight: $this.css('lineHeight')
30     });
31     $(this).keyup(update).blur(update).focus(update);
32     update.apply(this);
33 });
34 return this;
35 };
36 }(jQuery));

```

Listing 4.4: Wtyczka jQuery odpowiadająca za zmianę rozmiaru pól tekstowych

Innymi usprawnieniami interfejsu użytkownika w aplikacji są np.:

- Podpowiedzi pojawiające się po najechaniu myszką na część elementów strony. Aby ułatwić ich tworzenie, powstała wtyczka do jQuery. Dzięki jej użyciu aby dodać podpowiedź do dowolnego elementu HTML, wystarczyło nadać jej klasę CSS `tip` i treść podpowiedzi nadać za pomocą metody jQuery `.data('tip', 'Odpowiedni tekst')`.
- Okna dialogowe zastępujące standardowe funkcje typu `alert`, `confirm`, `prompt`. Pojawiające się w obrębie strony okno można dowolnie modyfikować.

#### 4.2.2. Parsowanie i drukowanie wikitekstu

Fundamentalne znaczenie dla aplikacji ma moduł odpowiadający za parsowanie wikitekstu — kodu źródłowego haseł. O trudności zagadnienia niech świadczy fakt, że przez ponad 10 lat istnienia Wikipedii nie udało się stworzyć edytora WYSIWYG — i prawdopodobnie w dalszym ciągu nie będzie to wykonalne ze względu na coraz większy poziom skomplikowania artykułów. Tylko dzięki wyjątkowo przejrzystej strukturze polskiego Wikisłownika udało się stworzyć aplikację. Warto zaznaczyć, że kilka lat temu także polska edycja internetowego słownika nie nadawała się do tego typu projektu — hasła udało się uprzątnąć za pomocą botów sterowanych przez kilka osób, które włożyły w to dużo wysiłku.

Najistotniejszą funkcją parsera jest podział hasła na sekcje językowe, a następnie na podsekcje — zgodnie z ogólnym schematem danego języka. Dzięki dobrej obsłudze wyrażeń regularnych w JavaScript kod odpowiadający za to jest dość krótki, jednak stworzenie prawidłowo działającej aplikacji wymagało długich testów, przeprowadzanych zarówno przez autora, jak i społeczność Wikisłownika.

O ile parsowanie hasła odbywa się według jasnych reguł, o tyle generowanie wikitekstu z poszczególnych fragmentów artykułu okazało się o wiele trudniejsze. Problemem nie było stworzenie kodu, który byłby poprawny, ale zadbanie o to, by aplikacja nie powodowała zbędnych zmian w kodzie. Kluczowa jest funkcja `EPrinter.recalculateCode`, zaprezentowana w wersji uproszczonej na listingu 4.5.

```
1 recalculateCode : function () {
2     var id, sec, i, j, subs,
3         code = [],
4         sortableSections = [];
5     ...
6     /* sortableSections zawiera spis tych sekcji, które mają znaleźć się w hasle */
7     /* kolejne kawałki kodu dodawane są do tablicy code, która na koniec jest łączona
   w jeden łańcuch znaków */
8     for (i in sortableSections) {
9         if (sortableSections.hasOwnProperty(i)) {
10            sec = sortableSections[i];
11            if (sec.id === EConstants.SECTION_ID_INTRO) {
12                /* sekcja wstępna */
13                /* EUi.val() pobiera zawartość pola tekstowego w formularzu */
14                code.push(EUi.val(EConstants.SECTION_ID_INTRO, '') + '\n');
15            } else {
16                /* tytuł sekcji */
17                code.push('== ' + sec.title + ' ==\n');
18                /* poszczególne podsekcje */
19                for (j = 0; j < sec.subsections.length; j += 1) {
20                    subs = sec.subsections[j];
21                    if (subs.active) {
22                        subs.content = EUi.val(sec.id, subs.title);
23
24                        if (!subs.title && subs.content) {
25                            /* podsekcja wstępna */
26                            code.push(subs.content + '\n');
27                        } else if (subs.title && !subs.content) {
28                            /* pusta podsekcja */
29                            code.push('{{' + subs.title + '}}\n');
30                        } else ... {
31                            code.push('{{' + subs.title + '}}' + EPrinter.
                                adequateWhitespace(subs) + subs.content + '\n');
32                        }
33                    }
34                }
35                code.push('\n');
36            }
37        }
38    }
39    /* łączenie w jeden napis i usunięcie wielokrotnych spacji */
40    return $.trim(code.join('')).replace(/ {2,}/g, ' ');
41 }
```

Listing 4.5: Funkcja `EPrinter.recalculateCode`

Ciekawą jest funkcja `EPrinter.adequateWhitespace`, użyta w powyższym kodzie w części odpowiadającej za wydrukowanie danej podsekcji. Jak się okazało, sposób drukowania poszczególnych podsekcji nie jest jednorodny. Za każdym razem istnieje kilka możliwości:

- Zawartość podsekcji zaczyna się od nowej linii.
- Zawartość podsekcji zaczyna się od spacji.
- Zawartość podsekcji zaczyna się od nowej linii i dwukropka ze spacją ( : ) powodującego wcięcie tekstu.

Aby ustalić, który wzorec wybrać, trzeba wziąć pod uwagę wiele warunków. Na listingu 4.6 przedstawiono funkcję to realizującą, opatrzoną dokładnymi komentarzami.

```
1 adequateWhitespace : function (subsection) {
2     var str = subsection.content;
3     /*
4      * Teksty zaczynające się od dwukropka, gwiazdki, zaczynające się od "<references
5      * ", "{litera|", "{kolor|", szablony zaczynające się na "{zch-", linki do
6      * grafiki (file:, grafika: image: media: plik:, to samo duża litera, możliwe
7      * białe znaki między nawiasami kwadratowymi a tym słowem),...
8      */
9     if (str.search(/[:\*#]|<references|\\{\\((litera|kolor)\\)|\\{\\{zch-|\\[\\(file|image|
10      grafika|plik|media):/i) === 0) {
11         return '\n';
12     }
13     /*
14      * ...teksty w polach "znaczenia", "przykłady" oraz "tłumaczenia" nie mogą
15      * występować zaraz po szablonie, jeśli występują, muszą być przeniesione bez
16      * dodawania dwukropka.
17      */
18     if (EConstants.SUBSECTIONS_WITH_NL.indexOf(subsection.title) !== -1) {
19         return '\n';
20     }
21     /*
22      * Inne teksty składające się z więcej niż jednej linii powinny być przeniesione z
23      * dodaniem dwukropka i spacji na początku pierwszej linii
24      */
25     if (str.indexOf('\n') !== -1 && str.search(/[:\*#]/) !== 0) {
26         return '\n: ';
27     }
28     /*
29      * Wpp: dla wypełnionych przed edycją pól zachowujemy istniejące formatowanie, o
30      * ile dane pole już było niepuste.
31      */
32     if (subsection.initcontent) {
33         return subsection.initmultiline ? '\n: ' : ' ';
34     }
35     /*
```

```

28      * w polach pustych przed edycja: w sekcjach "wymowa", "transliteracja", "
      transliteracja", "ortografie", "klucz", "kreski", "czytania", "hanja-kreski"
      defaultem jest pisanie bezposrednio po szablonie (po spacji)...
29      */
30      if (EConstants.SUBSECTIONS_WITHOUT_NL.indexOf(subsection.title) !== -1) {
31          return ' ';
32      }
33      /*
34      * a w pozostalych od nastepnej linii (jesli nie jest to "znaczenie" ani pierwsza
      sekcja ani "przyklady", ani "tlumaczenia", a tekst nie zaczyna sie od
      dwukropka lub gwiazdki, to program powinien sam dodac dwukropek i spacje)
35      */
36      return '\n: ';
37  }

```

Listing 4.6: Funkcja `EPrinter.adequateWhitespace`

Moduł `EPrinter` odpowiada także za generowanie kodu z danych pobranych w sposób automatyczny. Niektóre z tych funkcji zostały opisane w kolejnej sekcji.

### 4.3. Automatyzacja edycji hasła

Drugą funkcją obok uproszczenia ogólnego procesu edycji, jaką spełniać ma aplikacja, jest zautomatyzowanie niektórych działań wykonywanych przez redaktorów. W sekcji 2.4.2 wspomniano o dużym poziomie redundancji, będącym prostą konsekwencją rozbicia projektu na wersje językowe i niemożności zachowania pierwszej postaci normalnej danych w bazie Wikisłownika. Spora część danych wprowadzanych do hasła jest już obecna gdzie indziej — czy to w innych hasłach polskiego Wikisłownika, czy też w innych wersjach językowych. Wielu redaktorów wykorzystuje przykładowo inne edycje projektu do wprowadzania danych takich jak wymowa w międzynarodowym alfabecie fonetycznym (IPA), grafiki czy nagrania dźwiękowe, zaś na podstawie samej polskiej wersji możliwe jest chociażby kopiowanie przykładów użycia z jednego hasła do drugiego.

#### 4.3.1. API MediaWiki

Automatyzacja edytowania Wikisłownika (jak i innych projektów Fundacji) byłaby niezwykle trudna bez interfejsu programistycznego dołączonego do każdej witryny opartej na MediaWiki [20]. API umożliwia uproszczony dostęp do bazy Wikisłownika w licznych formatach wymiany danych takich jak XML, JSON, YAML czy WDDX. Aby otrzymać odpowiedź na dowolne zapytanie, wystarczy pobrać dane z adresu <http://pl.wiktionary.org/w/api.php> uzupełnionego o odpowiednie parametry.



Ponieważ analogiczną funkcję ma każdy projekt Wikimedia, możliwe jest zautomatyzowane pobieranie informacji nawet z kilkudziesięciu edycji językowych równocześnie. Przeszkodą w przetwarzaniu tak pobieranych danych jest *Same Origin Policy*, czyli koncepcja ograniczająca pobieranie danych za pomocą skryptów pomiędzy różnymi stronami internetowymi [41]. Z tego powodu niemożliwe jest wykorzystanie większości formatów, w tym XML. Aby obejść niektóre ograniczenia, konieczne jest zastosowanie tzw. JSONP (lub: JSON-P) [42]. Jest to sposób użycia istniejących technologii, który umożliwia pobieranie danych w JavaScript z innego serwera. Dane w formacie JSON uzupełniane są na zdalnym serwerze o wywołanie funkcji zdefiniowanej w parametrze, z którym skrypt został wywołany.

Istnieje kilka bibliotek, które mają ułatwiać operacje na API MediaWiki. Żadna z nich jednak nie spełnia swojej roli w odpowiedni sposób — utrudnione jest dokonywanie kilku zapytań równocześnie, biblioteki bywają też przeładowane funkcjami, które nie są potrzebne w projekcie dla Wikisłownika. Dlatego na jego potrzeby napisano od podstaw moduł EApi (w pliku `ed.api.js`), będący integralną częścią aplikacji. Najistotniejsze funkcje modułu przedstawiono na listingu 4.7.

```
1  EApi = {
2      /* Zwraca adres URL dla danego języka i projektu */
3      url : function (lang, project) {
4          if (lang === undefined) {
5              lang = 'pl';
6          }
7          if (project === undefined) {
8              project = EConstants.WIKTIONARY;
9          }
10         return 'http://' + lang + '.' + project + '.org/w/api.php?';
11     },
12
13     /* Wykonanie zapytania pod danym adresem. Do podanych pól dołączane są standardowe
14        */
15     ask__prv : function (query, url) {
16         if (url === undefined) {
17             url = EApi.url();
18         }
19         query.action = 'query';
20         query.format = 'json';
21         query.meta = 'siteinfo';
22         query.callback = 'EApi.callback';
23         url += $.param(query);
24         mw.loader.load(url);
25     },
26
27     /* Wykonanie pojedynczego zapytania */
28     ask : function (query, callback, url) {
```

```

28     if (EApi.waiting) {
29         jAlert(EStr.WAITING_FOR_API);
30         return -1;
31     }
32     EApi.waitingName = callback;
33     EApi.waiting = 1;
34     EApi.ask__prv(query, url);
35     return 0;
36 },
37
38 /* Wykonanie kilku zapytan naraz */
39 askMore : function (queries, callback) {
40     var i, count = 0;
41
42     if (EApi.waiting) {
43         jAlert(EStr.WAITING_FOR_API);
44         return -1;
45     }
46     EApi.waitingName = callback;
47
48     for (i in queries) {
49         if (queries.hasOwnProperty(i)) {
50             count += 1;
51         }
52     }
53     EApi.waiting = count;
54     $.each(queries, function (url, query) {
55         EApi.ask__prv(query, url);
56     });
57     return 0;
58 },
59
60 /* Funkcja wywoływana po nadejściu pojedynczego rezultatu. Wynik zapytania
61    dodawany jest do tablicy waitingResults. Jesli wszystkie rezultaty juz sie
62    pojawily, wykonywana jest funkcja o nazwie zapisanej przedtem w zmiennej
63    waitingName z argumentem waitingResults */
64 callback : function (res) {
65     var tmp = String(EApi.waitingName);
66     EApi.waitingResults.push(res);
67     EApi.waiting -= 1;
68     if (!EApi.waiting) {
69         EApi.waitingName = '';
70         EUtil.executeFn(tmp, window, EApi.waitingResults);
71         EApi.waitingResults = [];
72     }
73 },
74
75 waiting : 0,
76 waitingName : '',
77 waitingResults : []
78 };

```

Listing 4.7: Moduł EApi

Najprostsze użycie modułu EApi jest już nieskomplikowane: pozostaje zdefiniować dwie

funkcje, z których pierwsza odpowiada za przygotowanie i wysłanie zapytania do wybranych projektów, druga zaś za przetworzenie i wyświetlenie wyników. Wszystkie tego typu funkcje zebrano w module EAutomator. W przyszłości możliwe jest podzielenie tego modułu na mniejsze, z których każdy odpowiadałby konkretnemu zastosowaniu API MediaWiki. Funkcje przygotowane w ramach opisywanej aplikacji opisano bardziej szczegółowo na kolejnych stronach.

### 4.3.2. Funkcje automatyzujące edycję haseł

#### Aktualizacja interwiki

Linki interwiki w polskim Wikisłowniku aktualizowane są przez boty. Te działają sprawnie, jednak siłą rzeczy musi wystąpić opóźnienie między utworzeniem hasła a dodaniem interwiki. Czasem redaktorowi przydatna byłaby wiedza o innych wersjach językowych hasła, zanim linki do nich pojawią się w artykule. Poniżej podano parametry zapytania, które zazwyczaj odnajdzie wszystkie wersje językowe.

**Projekty**      Największe Wikisłowniki (angielski, hiszpański, francuski, niemiecki, rosyjski) oraz Wikisłowniki odpowiadające sekcjom językowym w hasle

**Zapytanie**

```
query = {  
    titles: mw.config.get('wgTitle'),  
    prop: 'langlinks',  
    lllimit: 200  
};
```

(Linki interwiki ze stron o tytule równemu tytułowi obecnej strony)

Poniżej przedstawiono kod odpowiadający za przetworzenie wyników i aktualizację pola tekstowego zawierającego linki interwiki. Tak wygląda ogólny schemat funkcji wywołanych po otrzymaniu rezultatów. W przypadku innych funkcji zamiast podawania całego kodu omówione zostaną najciekawsze problemy.

```
1 fillInterwikiRe : function (results) {  
2     var iwikiString, curIwiki, re,  
3         iwikis = [];  
4  
5     $.each(results, function () {  
6         var res = this;  
7         if (res.query === undefined || res.query.pages === undefined) {  
8             /* Nieprawidłowy wynik */
```

```

9         return false;
10    }
11    $.each(res.query.pages, function (j, val) {
12        if (j === '-1') {
13            /* Indeks strony == -1 oznacza brak strony w danym projekcie */
14            return false;
15        }
16        if (iwikis.indexOf(res.query.general.lang) === -1 && res.query.general.
            lang !== 'pl') {
17            /* Do interwiki dodajemy dany projekt */
18            iwikis.push(res.query.general.lang);
19        }
20        if (val.langlinks === undefined) {
21            return false;
22        }
23        $.each(val.langlinks, function () {
24            /* val.langlinks zawiera tytuły stron, do których w danym projekcie
                występują interwiki */
25            if (this['*'] === mw.config.get('wgTitle') && iwikis.indexOf(this.lang
                ) === -1 && this.lang !== 'pl') {
26                iwikis.push(this.lang);
27            }
28        });
29        return true;
30    });
31    return true;
32});
33/* sortowanie interwiki według ustalonego klucza */
34iwikis.sort(function (a, b) { return EConstants.INTERWIKI_ORDER.indexOf(a) -
    EConstants.INTERWIKI_ORDER.indexOf(b); });
35/* przygotowanie łańcucha znaków ze wszystkimi interwiki */
36iwikiString = $.map(iwikis, function (val) {
37    return '[' + val + ':' + mw.config.get('wgTitle') + ']]';
38}).join(' ');
39curIwiki = $('#ed_0000_').val();
40if (curIwiki === '') {
41    /* dotąd sekcja wstępna była pusta */
42    /* .autosize() powoduje dostosowanie wysokości pola */
43    $('#ed_0000_').val(iwikiString).autosize();
44} else {
45    /* zastąpienie poprzedniej wersji */
46    re = new RegExp('(\\[[\\[[a-z\\-]]+':':'+ mw.config.get('wgTitle') + '
        \\]\\]\\s*)+');
47    $('#ed_0000_').val($.trim(iwikiString + curIwiki.replace(re, '\n'))).
        autosize();
48}
49EApi.done(EConstants.MODE_IW);
50}

```

Listing 4.8: Funkcja EAutomator.fillInterwikiRe

## Pobieranie zapisu wymowy w IPA

Zapis wymowy w międzynarodowym alfabecie fonetycznym (IPA) to jedna z tych informacji, które można bez problemu przenosić pomiędzy różnymi wersjami językowymi Wikisłownika. Często zdarza się, że w wielu wersjach językowych zapis pewnego słowa jest podany, brakuje go zaś w polskiej edycji. Najczęściej IPA nie jest ręcznie wpisywane do hasła, ale kopiowane z innego źródła — jest to oczywiście wygodniejsze z powodu konieczności używania innego alfabetu. Aby ułatwić wstawienie tej informacji do hasła, można poprzez API pobrać zawartość haseł z interwiki. Ważne jest, by dobrze sparsować otrzymane hasła w poszukiwaniu IPA.

**Projekty** Wikisłowniki odpowiadające sekcjom językowym w hasle i inne z linków interwiki

**Zapytanie**

```
query = {  
    titles: mw.config.get('wgTitle'),  
    prop: 'revisions',  
    rvprop: 'content'  
};
```

(Zawartość stron o tytule równemu tytułowi obecnej strony)

Hasła w innych wersjach językowych mogą dotyczyć wielu języków, w których występuje dane słowo. Koniecznie trzeba zminimalizować więc ryzyko wstawienia wymowy wyrazu w innym języku niż w obecnie edytowanej sekcji przez niedoświadczonego użytkownika. Przy okazji implementacji funkcji odpowiadającej za przetworzenie wyników okazało się, że programistyczny podział hasła na sekcje w innej wersji językowej wcale nie musi być tak prosty jak w wersji polskiej. Choć wspólne dla projektów jest stosowanie nagłówków dla poszczególnych języków, to w części wersji nagłówki te są generowane dynamicznie przez szablony. Przykładowo nagłówek sekcji języka polskiego w Wikisłowniku hiszpańskim ma postać `{{PL-ES|Nazwa}}`, a w Wikisłowniku francuskim: `== {{=pl=}} ==`.

Wydaje się, że dobrym rozwiązaniem problemu jest podanie przy każdej wykrytej informacji o IPA w obcojęzycznej wersji hasła także nagłówek sekcji, w której ją znaleziono. Co prawda istnieją Wikisłowniki takie jak wietnamski, gdzie nie sposób wyodrębnić poszczególne sekcje, jednak dla większości udało się to osiągnąć. W ten sposób redaktor otrzymuje zawartość nagłówka, która pozwala się zorientować, o jaki język może chodzić. Listing 4.9

przedstawia fragment funkcji, która przy użyciu wyrażeń regularnych pozwala na podział artykułu na sekcje. Znacznikami <BE> i <EN> oznaczane są granice nagłówków i na tej podstawie kod jest dzielony na sekcje w dalszej części funkcji.

```

1  if (lang === undefined) {
2      /* Polski: naglowki 2. stopnia */
3      code = code.replace(/^==(^[^=][^\\n]+?)==\\s*$/gm, '<BE>$1<EN>');
4  } else {
5      switch (lang) {
6          case 'ru':
7              /* Rosyjski: naglowki 1. stopnia */
8              code = code.replace(/^==(^[^=][^\\n]+?)=)\\s*$/gm, '<BE>$1<EN>');
9              break;
10         case 'fr':
11         case 'li':
12         case 'nl':
13         case 'oc':
14             /* Francuski i inne: szablony typu {{=en=}} w naglowkach */
15             code = code.replace(/^((==\\s*)?\\{\\{([^=\\-][^\\n]*)?=\\}\\}\\(\\s*==?)\\}\\s*$/gm, '<BE
16                 >$1<EN>');
17             break;
18         case 'lv':
19             /* Lotewski: szablony typu {{-en-}} */
20             code = code.replace(/^\\{\\{-([^=\\-][^\\n]*)-\\}\\}\\}\\s*$/gm, '<BE>$1<EN>');
21             break;
22         case 'co':
23         case 'ga':
24             code = code.replace(/^\\{\\{-\\w\\w-\\}\\}\\}\\s*$/gm, '<BE>$1<EN>');
25             break;
26         case 'is':
27             code = code.replace(/^\\{\\{-\\w{2,3}-\\}\\}\\}\\s*$/gm, '<BE>$1<EN>');
28             break;
29         case 'it':
30             /* Wloski: szablony typu {{in|en|art}} */
31             code = code.replace(/^\\{\\{in|[^\\}\\}\\}\\}\\}\\s*$/gm, '<BE>$1<EN>');
32             break;
33         case 'es':
34             /* Hiszpanski: szablony typu {{FR-ES|a}} */
35             code = code.replace(/^\\{\\{[A-Z\\-]{2,}\\}\\}\\}\\}\\s*$/gm, '<BE>$1<EN>');
36             break;
37         case 'af':
38             /* Afrykanerski: dwa rozne sposoby */
39             code = code.replace(/^\\{\\{-\\w\\w-\\}\\}\\}\\s*$/gm, '<BE>$1<EN>');
40             code = code.replace(/^==(^[^=][^\\n]+?)==)\\s*$/gm, '<BE>$1<EN>');
41             break;
42         default:
43             code = code.replace(/^==(^[^=][^\\n]+?)==)\\s*$/gm, '<BE>$1<EN>');
44             break;
45     }

```

Listing 4.9: Fragment funkcji EParser.getSections

Ponieważ w większości wersji językowych zapisy IPA podawane są w określonym szablonie, przy użyciu funkcji pomocniczych odnajdujących parametry szablonów udało się doprowadzić do pobierania IPA z kilkunastu Wikisłowników. Listing 4.10 pokazuje, jak różnorodne są sposoby podawania IPA.

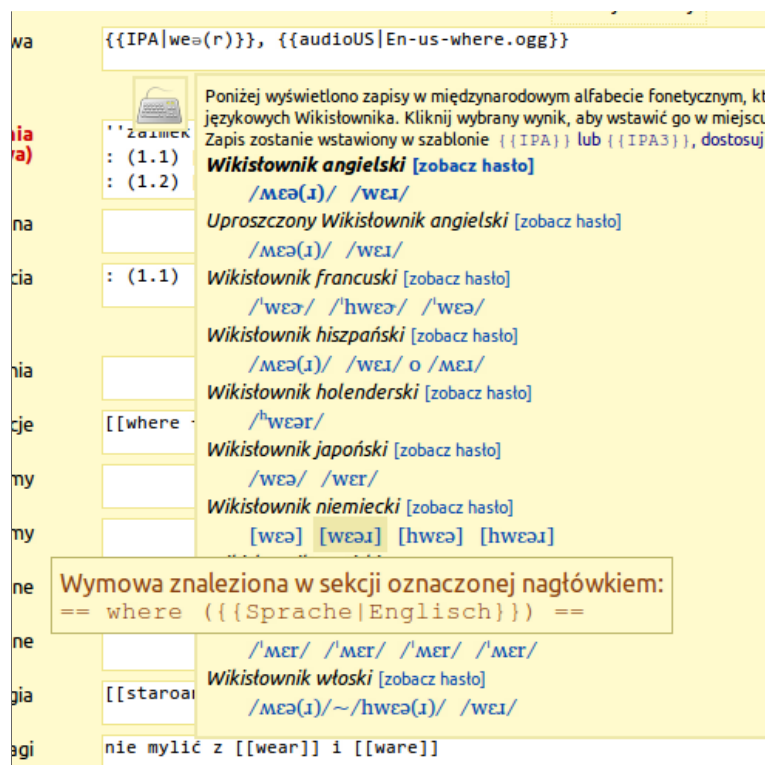
```
1  extractIPA : function (str, lang) {
2      switch (lang) {
3          case 'en': case 'cs': case 'sk': case 'it': case 'af': case 'ko': case 'nl':
4          case 'co': case 'vi': case 'ga': case 'is': case 'li': case 'lv': case 'no':
5          case 'sl': case 'ja': case 'simple':
6              return EAutomator.extractFirstArgsFromTemplates(str, 'IPA', lang);
7          case 'fr': case 'mg': case 'oc':
8              return EAutomator.extractFirstArgsFromTemplates(str, 'pron', lang);
9          case 'de':
10             return EAutomator.extractFirstArgsFromTemplates(str, 'Lautschrift', lang);
11          case 'es':
12             return EAutomator.extractAllArgsFromTemplates(str, 'pronunciación', lang);
13          case 'ca':
14             return EAutomator.extractSecondArgsFromTemplates(str, 'pron', lang);
15          case 'ro':
16             return EAutomator.extractFirstArgsFromTemplates(str, 'AFI', lang);
17          case 'et':
18             return EAutomator.extractFirstArgsFromTemplates(str, 'hääldus', lang);
19          case 'el':
20             return EAutomator.extractFirstArgsFromTemplates(str, 'ΔΦΑ', lang);
21          case 'eo':
22             return EAutomator.extractFirstArgsFromTemplates(str, 'IFA', lang);
23          case 'tl':
24             return EAutomator.extractFirstArgsFromTemplates(str, 'API', lang);
25          case 'ru':
26             return EAutomator.extractIPA_ru(str);
27          default:
28             return [];
29      }
30 }
```

Listing 4.10: Funkcja EAutomator.extractIPA

Dodatkową komplikacją jest występowanie IPA w dwóch wariantach. Zapis w nawiasach kwadratowych to zapis fonetyczny, natomiast między ukośnikami — zapis fonologiczny. W różnych wersjach językowych używane są różne warianty, natomiast w polskiej edycji podawane są oba. Aplikacja bierze pod uwagę, jak zbudowany jest szablon w danym Wikisłowniku, i w zależności od tego pozwala na wstawienie IPA w szablonie {{IPA}} lub {{IPA3}}.

Za wyświetlenie zapisów IPA, które można następnie wstawić jednym kliknięciem, odpowiedzialna jest funkcja EPrinter.ipaResult. Jej efektem jest pojawienie się okienka,

z którego można wybrać odpowiedni zapis (patrz ilustracja 4.4). Nagłówki sekcji językowych, z których pochodzą IPA, wyświetlane są po najechnaniu myszką.



Ilustracja 4.4: Fragment okna wyświetlającego pobrane zapisy IPA

## Pobieranie grafik

Innym elementem kopiowanym pomiędzy wersjami językowymi są ilustracje do haseł. Jest oczywiste, że ilustracja danego wyrazu będzie równie dobra w każdej edycji Wikisłownika. Ponieważ znakomita większość używanych w projektach Wikimedia grafik znajduje się na serwerach Wikimedia Commons (centralnego repozytorium plików dostępnych na wolnej licencji), a w polskich projektach tylko takie grafiki są używane, możliwe jest automatyczne odnajdywanie tych ilustracji.

**Projekty** Wikisłowniki odpowiadające sekcjom językowym w haśle i inne z linków interwiki

**Zapytanie**

```
query = {
    titles: mw.config.get('wgTitle'),
    prop: 'revisions',
    rvprop: 'content'
};
```



(Zawartość stron o tytule równemu tytułowi obecnej strony)

Skrypt odpowiadający za przetworzenie wyników zapytania działa dwuetapowo. Na początku z obcojęzycznych haseł ekstrahowane są nazwy plików graficznych — jest to dość proste do wykonania, ponieważ standardowy sposób wstawiania ich do haseł sprawia, że zawsze poprzedzone są dwukropkiem (alternatywnie mogą być parametrami szablonów, a więc występować po znakach `|` lub `=`), kończą się zaś oczywiście rozszerzeniem formatu graficznego. Już po odnalezieniu tych nazw możliwe byłoby zakończenie działania funkcji i umożliwienie wprowadzenia redaktorowi pliku do polskiego hasła na podstawie jego nazwy. Aby jednak zwiększyć intuicyjność obsługi aplikacji i wyświetlić użytkownikowi miniaturkę każdego z plików, wykonywane jest kolejne zapytanie:



Ilustracja 4.5: Fragment okna do wyboru ilustracji wraz ze wskazaną miniaturą

**Projekty**      Wikimedia Commons

**Zapytanie**

```
query = {  
  titles: allImages.join('|'),  
  prop: 'imageinfo',  
  iiprop: 'url',  
  iiurlwidth: 150,  
  iiurlheight: 150  
};
```

(Adresy URL miniatur o rozmiarach 150 × 150 px dla podanych plików)

Lokalizacja miniatury ilustracji o zadanej wielkości nie jest bezpośrednio zależna od nazwy pliku — MediaWiki używa funkcji haszujących, aby umieścić ją w jednym z wielu katalogów. Dlatego zwykłymi metodami niezależnymi od MediaWiki nie jest możliwe natychmiastowe wyświetlenie ilustracji bez zapytania do API Wikimedia Commons, które przekaże niezbędne dane. Po otrzymaniu tej informacji aplikacja ładuje odpowiednie obrazy i wyświetla je po najechaniu myszą na nazwę pliku, jak pokazano na ilustracji 4.5.

### Pobieranie nagrań wymowy

Aplikacja umożliwia także automatyczne odnajdywanie plików dźwiękowych z wzorcową wymową danego słowa. Wyszukiwanie to odbywa się w podobny sposób jak w przypadku ilustracji — tutaj kod haseł w innych językach przeszukiwany jest na okoliczność wystąpienia nazwy pliku zakończonego na .ogg. Zapytanie o zawartość stron ma postać identyczną jak w poprzednim przykładzie.

Nagrania wymowy, podobnie jak ilustracje, umieszczane są w Wikimedia Commons. Dzięki zasadom obowiązujących w tym projekcie nazwy tych plików są zstandaryzowane: mają postać <ISO>-<wyraz>.ogg (np. De-Meerschweinchen.ogg dla słowa *Meerschweinchen* — *świnka morska*). Z tego powodu możliwe jest dodatkowe wyszukanie pliku bezpośrednio w zasobach Commons za pomocą API tego projektu. Dodatkowo dla słów angielskich i niemieckich często pojawiają się warianty wymowy amerykańskiej, australijskiej czy austriackiej. Listing 4.11 pokazuje, jak konstruowane jest zapytanie do Commons.

```
1 lang = $.ucFirst(EUtil.getActiveLangCode());
2 titles.push('File:' + lang + '-' + mw.config.get('wgTitle') + '.ogg');
3 if (lang === 'En') {
4     titles.push('File:' + lang + '-uk-' + mw.config.get('wgTitle') + '.ogg');
5     titles.push('File:' + lang + '-us-' + mw.config.get('wgTitle') + '.ogg');
6     titles.push('File:' + lang + '-au-' + mw.config.get('wgTitle') + '.ogg');
7 }
8 if (lang === 'De') {
9     titles.push('File:' + lang + '-at-' + mw.config.get('wgTitle') + '.ogg');
10 }
11
12 queries[EApi.commonUrl()] = { titles: titles.join('|'), prop: 'info' };
```

Listing 4.11: Fragment funkcji EAutomator.getAudio

Pobieranie przykładów użycia

#### **4.4. Wdrożenie i dalszy rozwój**

...



## **Rozdział 5**

# **Podsumowanie**



## Dodatek A

# Zawartość płyty CD

1. Niniejsza praca:
  - w formacie PDF,
  - kod źródłowy  $\text{\LaTeX}$ ,
2. Kod źródłowy aplikacji:
  - w postaci modułowej,
  - w postaci złączonej do jednego pliku,
3. Skrypt do łączenia aplikacji w jeden plik.





## **Dodatek B**

### **Wygląd aplikacji — zrzuty ekranu**



# Bibliografia

- [1] Stowarzyszenie Wikimedia Polska. Fundacja Wikimedia. [http://pl.wikimedia.org/w/index.php?title=Fundacja\\_Wikimedia&oldid=10461](http://pl.wikimedia.org/w/index.php?title=Fundacja_Wikimedia&oldid=10461). [Dostęp 6 sierpnia 2011].
- [2] List of Wikipedias — Meta, discussion about Wikimedia projects. [http://meta.wikimedia.org/w/index.php?title=List\\_of\\_Wikipedias&oldid=2715916](http://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias&oldid=2715916). [Dostęp 6 sierpnia 2011].
- [3] Wikimedia Foundation. <http://www.wikimedia.org/>. [Dostęp 5 sierpnia 2011].
- [4] Licensing update/Result — Meta, discussion about Wikimedia projects. [http://meta.wikimedia.org/w/index.php?title=Licensing\\_update/Result&oldid=1776040](http://meta.wikimedia.org/w/index.php?title=Licensing_update/Result&oldid=1776040), 2009. [Dostęp 5 sierpnia 2011].
- [5] Help:Contents — MediaWiki, The Free Wiki Engine. <http://www.mediawiki.org/w/index.php?title=Help:Contents&oldid=421744>, 2011. [Dostęp 5 sierpnia 2011].
- [6] Dlaczego Wikisłownik? — Wikisłownik, wolny wielojęzyczny słownik. [http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Dla\\_czego\\_Wikis%C5%82ownik&oldid=2292379](http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Dla_czego_Wikis%C5%82ownik&oldid=2292379), 2007. [Dostęp 5 sierpnia 2011].
- [7] Administrators — Wiktionary. <http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Administratorzy&oldid=1316722>, 2003. [Dostęp 5 sierpnia 2011].
- [8] Administratorzy — Wikisłownik, wolny wielojęzyczny słownik. <http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Administratorzy&oldid=1316722>, 2004. [Dostęp 5 sierpnia 2011].
- [9] Administrators of Wikimedia projects/Wikipedias — Meta, discussion about Wikimedia projects. [http://meta.wikimedia.org/w/index.php?title=Administrators\\_of\\_Wikimedia\\_projects/Wikipedias&oldid=2748948](http://meta.wikimedia.org/w/index.php?title=Administrators_of_Wikimedia_projects/Wikipedias&oldid=2748948), 2011. Dostęp 5 sierpnia 2011.

- [10] Strona główna — Wikisłownik, wolny wielojęzyczny słownik. [http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Strona\\_g%C5%82%C3%B3wna&oldid=2298705](http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Strona_g%C5%82%C3%B3wna&oldid=2298705), 2004. [Dostęp 5 sierpnia 2011].
- [11] Wiktionary: List of Wiktionaries — Meta, discussion about Wikimedia projects. [http://meta.wikimedia.org/w/index.php?title=Wiktionary&oldid=2424480#List\\_of\\_Wiktionaries](http://meta.wikimedia.org/w/index.php?title=Wiktionary&oldid=2424480#List_of_Wiktionaries). [Dostęp 5 sierpnia 2011].
- [12] Zasady tworzenia haseł — Wikisłownik, wolny wielojęzyczny słownik. [http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Zasady\\_tworzenia\\_hase%C5%82&oldid=2294171](http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Zasady_tworzenia_hase%C5%82&oldid=2294171), 2004. [Dostęp 5 sierpnia 2011].
- [13] O. Matyja. Wikipedysta:Olafbot — Wikisłownik, wolny wielojęzyczny słownik. <http://pl.wiktionary.org/w/index.php?title=Wikipedysta:Olafbot&oldid=2123349>. [Dostęp 6 sierpnia 2011].
- [14] D. Ostaszewska, J. Tambor. *Fonetyka i fonologia współczesnego języka polskiego*. Wydawnictwo Naukowe PWN, Warszawa, wydanie 2, dodruk 4, sierpień 2009.
- [15] C.J. Date. *An Introduction to Database Systems*. Addison-Wesley, Boston, wydanie 8, 2004.
- [16] R. Glott, Ph. Schmidt, R. Ghosh. Wikipedia Survey — Overview of Results. Raport instytutowy, UNU-MERIT, Maastricht, 15 marca 2010. Dostępny w sieci: [http://www.wikipediastudy.org/docs/Wikipedia\\_Overview\\_15March2010-FINAL.pdf](http://www.wikipediastudy.org/docs/Wikipedia_Overview_15March2010-FINAL.pdf) [Dostęp 5 sierpnia 2011].
- [17] User:Cacycle/wikEd — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/w/index.php?title=User:Cacycle/wikEd&oldid=442519985>, 2011. [Dostęp 5 sierpnia 2011].
- [18] R. S. Geiger. The lives of bots. *Critical Point of View. A Wikipedia Reader*, strony 78–93. Institute of Network Cultures, Amsterdam, 2011. Dostępny w sieci: [http://www.networkcultures.org/\\_uploads/%237reader\\_Wikipedia.pdf](http://www.networkcultures.org/_uploads/%237reader_Wikipedia.pdf) [Dostęp 6 sierpnia 2011].
- [19] Wikisłownik:Boty — Wikisłownik, wolny wielojęzyczny słownik. <http://pl.wiktionary.org/w/index.php?title=Wikis%C5%82ownik:Boty&oldid=2020877>. [Dostęp 6 sierpnia 2011].
- [20] API:Main Page — MediaWiki, The Free Wiki Engine. [http://www.mediawiki.org/w/index.php?title=API:Main\\_page&oldid=407852](http://www.mediawiki.org/w/index.php?title=API:Main_page&oldid=407852). [Dostęp 6 sierpnia 2011].

- [21] M. A. Dennis. *wiki*. Encyclopædia Britannica. Encyclopædia Britannica Online. <http://www.britannica.com/EBchecked/topic/1192819/wiki>, 2011. [Dostęp 27 sierpnia 2011].
- [22] S. Vaknin. *Wikipedia, Criticism of*. <http://knol.google.com/k/wikipedia-criticism-of>, 2008. [Dostęp 27 sierpnia 2011].
- [23] J. Giles. Internet encyclopaedias go head to head. *Nature*, 438:900–901, 2005. doi: 10.1038/438900a.
- [24] G. E. Pataki, J. T. Dillon, M. McCormack. *New York State Project Management Guidebook*. New York State Office for Technology, Nowy Jork, wydanie 2, 2003. Dostępne online: <http://www.cio.ny.gov/pmmp/guidebook2/index.htm>.
- [25] Manual:Extensions — MediaWiki, The Free Wiki Engine. <http://www.mediawiki.org/w/index.php?title=Manual:Extensions&oldid=429069>. [Dostęp 28 sierpnia 2011].
- [26] Wikipedia:Technik/Skin/JS — Wikipedia. <http://de.wikipedia.org/w/index.php?title=Wikipedia:Technik/Skin/JS&oldid=92936675>. [Dostęp 28 sierpnia 2011].
- [27] Release notes/1.17 — MediaWiki, The Free Wiki Engine. [http://www.mediawiki.org/w/index.php?title=Release\\_notes/1.17&oldid=423029](http://www.mediawiki.org/w/index.php?title=Release_notes/1.17&oldid=423029), 2011. [Dostęp 29 sierpnia 2011].
- [28] ResourceLoader/Migration guide for extension developers — MediaWiki, The Free Wiki Engine. [http://www.mediawiki.org/w/index.php?title=ResourceLoader/Migration\\_guide\\_for\\_extension\\_developers&oldid=428906](http://www.mediawiki.org/w/index.php?title=ResourceLoader/Migration_guide_for_extension_developers&oldid=428906), 2011. [Dostęp 29 sierpnia 2011].
- [29] JQuery — MediaWiki, The Free Wiki Engine. <http://www.mediawiki.org/w/index.php?title=JQuery&oldid=420417>. [Dostęp 29 sierpnia 2011].
- [30] B. Bibeault, Y. Katz. *jQuery in Action*. Manning Publications, Greenwich, wydanie 2, 2010.
- [31] Documentation. Main Page — jQuery JavaScript Library. [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page). [Dostęp 29 sierpnia 2011].
- [32] Dokumentacja Geany. <http://www.geany.org/Documentation/Manual>. [Dostęp 29 sierpnia 2011].
- [33] Dokumentacja Firebuga. [http://getfirebug.com/wiki/index.php/Main\\_Page](http://getfirebug.com/wiki/index.php/Main_Page). [Dostęp 29 sierpnia 2011].
- [34] Dokumentacja GitHub. <http://help.github.com/>. [Dostęp 29 sierpnia 2011].

- [35] Repozytorium kdudzik/wikt — GitHub. <https://github.com/kdudzik/wikt>. [Dostęp 29 sierpnia 2011].
- [36] Dokumentacja The Regex Coach. <http://weitz.de/regex-coach/>. [Dostęp 29 sierpnia 2011].
- [37] Dokumentacja JSHint. <http://jshint.com/>. [Dostęp 29 sierpnia 2011].
- [38] Dokumentacja JavaScriptLint. <http://www.javascriptlint.com/docs/index.htm>. [Dostęp 29 sierpnia 2011].
- [39] Dokumentacja JSLint. <http://www.jslint.com/lint.html>. [Dostęp 29 sierpnia 2011].
- [40] Manual:Coding conventions — MediaWiki, The Free Wiki Engine. [http://www.mediawiki.org/w/index.php?title=Manual:Coding\\_conventions&oldid=429647](http://www.mediawiki.org/w/index.php?title=Manual:Coding_conventions&oldid=429647). [Dostęp 30 sierpnia 2011].
- [41] Mozilla Developer Network. Same origin policy for JavaScript. [https://developer.mozilla.org/en/Same\\_origin\\_policy\\_for\\_JavaScript](https://developer.mozilla.org/en/Same_origin_policy_for_JavaScript). [Dostęp 9 września 2011].
- [42] Defining Safer JSON-P. <http://www.json-p.org/>. [Dostęp 9 września 2011].

# Spis ilustracji

2.1. Polska edycja Wikipedii . . . . .	8
2.2. Polska edycja Wikisłownika . . . . .	12
2.3. Hasło w polskim Wikisłowniku . . . . .	15
2.4. Edycja hasła w polskim Wikisłowniku . . . . .	16
2.5. Próba stworzenia nowego hasła w polskim Wikisłowniku . . . . .	27
4.1. Kluczowe elementy interfejsu w trybie edycji . . . . .	45
4.2. Fragment nowego formularza w haśle z jedną sekcją językową . . . . .	47
4.3. Użycie klawiaturki ekranowej w nowym formularzu . . . . .	49
4.4. Fragment okna wyświetlającego pobrane zapisy IPA . . . . .	62
4.5. Fragment okna do wyboru ilustracji wraz ze wskazaną miniaturą . . . . .	63

# Spis tabel

2.1. Porównanie HTML i wikitekstu . . . . .	11
2.2. Porównanie Wikipedii i Wikisłownika — aspekty techniczne . . . . .	30
3.1. Porównanie Wikipedii i Wikisłownika — aspekty społecznościowe . . . . .	33

# Spis listingów

4.1. Ogólna struktura modułu JavaScript . . . . .	43
4.2. Funkcja EUi.rebindFormActions . . . . .	46
4.3. Moduł ESpecialChars . . . . .	48
4.4. Wtyczka jQuery odpowiadająca za zmianę rozmiaru pól tekstowych . . . . .	50
4.5. Funkcja EPrinter.recalculateCode . . . . .	52
4.6. Funkcja EPrinter.adequateWhitespace . . . . .	53
4.7. Moduł EApi . . . . .	55
4.8. Funkcja EAutomator.fillInterwikiRe . . . . .	57
4.9. Fragment funkcji EParser.getSections . . . . .	60
4.10. Funkcja EAutomator.extractIPA . . . . .	61
4.11. Fragment funkcji EAutomator.getAudio . . . . .	64