

УДК 658.012.01156

І.М. Білятинська<sup>1</sup><sup>1</sup>Український мовно-інформаційний фонд НАУ, м. Київ, Україна, ira121191@ukr.net

## СИСТЕМОТЕХНІЧНІ ЗАСАДИ СТВОРЕННЯ ON-LINE ВЕРСІЇ СЛОВНИКА УКРАЇНСЬКОЇ МОВИ

У статті розглянуто системотехнічні засади створення on-line версії Словника української мови. Виділено основні вимоги до засобів представлення лексикографічного матеріалу в мережі Інтернет. Окреслено коло завдань із розробки відповідного програмного забезпечення та технології й підходи, використані в процесі проектування.

СЛОВНИК УКРАЇНСЬКОЇ МОВИ, ЛЕКСИКОГРАФІЧНА СИСТЕМА, СИСТЕМОТЕХНІКА, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ПЛАТФОРМА РОЗРОБКИ

### Вступ

Мережа Інтернет за останні десятиліття перетворилася із засобу обміну відносно невеликими об'ємами даних до універсального технологічно-комунікаційного середовища, в якому розвивається інформаційне суспільство. Завдяки зниженню вартості мобільних пристроїв, комп'ютерної техніки та технологій їх підключення до Всесвітньої павутини із паралельним збільшенням швидкості передачі даних популярність Мережі зростає кожного дня, спонукаючи до розміщення величезної кількості інформаційних ресурсів і послуг on-line. За таких умов стає актуальним питання представлення в мережі Інтернет структурованої лінгвістичної інформації у вигляді електронних словників, або, загальніше, лексикографічних даних [1-4]. При цьому принциповими ми вважаємо два аспекти: а) якомога адекватніше й повне подання в таких системах структури і властивостей відповідних лексикографічних систем, що експлікують різноманітні мовні явища і б) забезпечення необхідного онлайн-ового функціоналу, системотехніка якого враховує різні режими функціонування системи (інструментальність, інтероперабельність, ергономічність тощо).

Керуючись цими настановами, в Українському мовно-інформаційному фонді Національної академії наук України реалізується проект «Словник української мови», в межах якого створено on-line версію Словника української мови (далі Система) для реалізації можливості доступу широкого кола користувачів до добре структурованого і великого за обсягом лексикографічного матеріалу. Щоб забезпечити якомога ширше охоплення аудиторії, при проектуванні цієї онлайн-ової системи було поставлено такі завдання:

- забезпечити подання матеріалу словника в зручному для користувача вигляді з широкими інтерфейсними можливостями;
- максимально використовувати відкриті стандарти;
- забезпечити роботу Системи в різних операційних середовищах (Windows, Linux, Android, iOS...) та на широкій апаратній базі (стаціонарні комп'ютери, планшети, смартфони);
- забезпечити прийнятний рівень користувацького інтерфейсу (usability);
- реалізувати різні адаптивні шаблони користувацького інтерфейсу для різних типів екранів з різною роздільною здатністю;
- забезпечити прийнятний рівень часу відповіді системи та можливість обслуговування великої кількості користувачів.

### 1. Структуризація схемотехнічних засад

Для забезпечення мобільності зазначеної мовно-інформаційної системи було вирішено використовувати концепцію проектування програмного забезпечення як сервісу SOA [1] та підходу SaaS [4].

Сервісно-орієнтована архітектура (Service-oriented architecture, SOA) – форма технологічної архітектури яка може складатися із комбінацій технологій, продуктів що підтримують розширення інфраструктури та слабо пов'язаних замінних компонентів, оснащених стандартизованими інтерфейсами для взаємодії за стандартизованими протоколам [1, с. 38].

До основних переваг використання цієї архітектури можна віднести:

- стандартизований підхід – сервіси базуються на загальноприйнятих стандартах, не прив'язуючись до конкретного програмного продукту, конкретної програмної платформи, тощо;
- слабке зв'язування – сервіси незалежні один від одного та орієнтовані на виконання вузького кола строго окреслених функцій;
- автономність;
- можливість повторного використання;
- відкритість.

SaaS (Software as a Service) – модель розгортання та реалізації програмного забезпечення, при якій користувач отримує як сервіс через мережу Інтернет завершений додаток, може налаштовувати деякі специфічні для застосунку параметри, а всією інфраструктурою, логікою програми, її розгортанням керує постачальник послуги [4, с. 51]. SaaS разом з IaaS (надання апаратної IT-інфраструктури, що включає сервери, мережі, пристрої зберігання інформації), та PaaS (апаратна IT-інфраструктура, операційна система та її API) репрезентують різні рівні моделі Cloud Computing. Зі свого боку, Cloud Computing (хмарні обчислення) – це стиль розробки і використання програмного забезпечення та обчислювальних ресурсів, в якому кінцеві продукти надаються у користування як сервіси через мережу Інтернет [4, 58].

Використання моделі SaaS при проектуванні Системи дозволило готовому продукту акумулювати наступні позитивні властивості:

- здійснення доступу до застосунку без необхідності встановлення додаткового програмного забезпечення на робоче місце користувача;
- доступ за допомогою звичайного веб-браузера;
- відсутність затрат налаштування та встановлення системи на користувацькі комп'ютери;

- популярність серед сучасних користувачів веб-сервісів;
- кросплатформеність та можливість використання на мобільних пристроях;
- використання вільного програмного забезпечення;
- централізоване обслуговування сервісу;
- експлуатація не потребує спеціальних навичок для роботи з пристроєм та прав адміністратора.

On-line версію Словника української мови створено в середовищі розробки Microsoft Visual 2013 Professional Edition з використанням базового фреймворку .Net framework 4.5.1. Ця платформа була обрана завдяки наявності ієрархічної множини бібліотек класів, засобів створення багаторівневих застосунків, підтримки використання різних мов програмування.

Робота з базою даних реалізована із застосуванням об'єктно-орієнтованої технології доступу до даних Entity framework 6.1.2. Платформа Entity Framework є набором технологій ADO.NET, які забезпечують розробку застосунків, що пов'язані з обробкою даних. Архітекторам і розробникам застосунків, що орієнтовані на обробку даних, доводиться враховувати необхідність досягнення двох цілком різних цілей. З одного боку, вони повинні моделювати сутності, зв'язки і логіку бізнес-задач, що розв'язуються, а з іншого – працювати з ядрами СУБД, що використовуються для збереження і отримання даних. Дані можуть розподілятися за кількома системами зберігання даних, в кожній з яких застосовуються свої протоколи, але навіть у додатках, що працюють з однією системою зберігання даних, необхідно підтримувати баланс між вимогами системи зберігання даних і вимогами написання ефективного і зручного для обслуговування коду програми.

Entity Framework дозволяє працювати з даними у формі специфічних для домену об'єктів і властивостей, таких як клієнти та їх адреси, без необхідності звертатися до базових таблиць і стовпців бази даних, де зберігаються ці дані; надає розробникам можливість працювати з даними на більш високому рівні абстракції, створювати і супроводжувати орієнтовані на дані застосунки, використовуючи менші обсяги коду, ніж у традиційних застосунках.

Для створення веб-застосунків, логіки застосування було використано фреймворк для розробки від Microsoft ASP.NET MVC 5.2.2, який поєднує в собі ефективність архітектури MVC та можливості платформи ASP.NET.

MVC (Model-view-controller) – схема використання декількох шаблонів проектування, за допомогою яких кінцевий програмний продукт розділений на три окремих компоненти: модель застосунку, користувацький інтерфейс (представлення) та логіка взаємодії [5, с. 46]. Даний патерн широко використовується під час розробки веб-застосунків завдяки наступним перевагам:

- взаємодія користувача з MVC-застосунком здійснюється відповідно природньому циклу: користувач здійснює дію, на яку застосунок реагує зміною моделі даних та наданням користувачу оновленого стану системи, що, безперечно, дуже зручно

для веб-застосунків, які надаються у вигляді серії HTTP-запитів та відповідей;

- зазначений підхід дозволяє використовувати різні технології розробки програмного забезпечення одночасно, які поєднуються між собою та можуть взаємодіяти;

- використання MVC-патерну забезпечує відокремлення представлення від даних та логіки їх опрацювання, тим самим забезпечуючи реалізацію трирівневої архітектури «представлення-логіка-дані».

В ASP.NET MVC-патерн реалізується за рахунок взаємодії трьох компонентів: контролерів, моделі та представлення. Модель являє собою опис логіки даних. У нашому випадку розроблення моделі словника базувалося на шаблоні Domain model, відповідно до якого створюється модель предметної області (доменна модель), що являє собою мережу взаємопов'язаних об'єктів, де кожен із них представляє собою певну значущу індивідуальність та описується множиною атрибутів. В моделі предметної області об'єднуються дані та процеси, існує складна мережа асоціацій та використовується наслідування [3].

Доменну модель було створено на основі вже готової лексикографічної бази даних Словника української мови і реалізовано у вигляді сукупності C# класів моделі даних (Entity Data Model) за допомогою технології роботи з даними Entity Framework. Контролери приймають запити та опрацьовують користувацьке введення, здійснюють взаємодію із моделлю, повертають користувачам результати обробки запитів. Для обміну даними між сервером та клієнтом використовуються API-контролери, які є частиною інтерфейсу ASP.NET WebApi 2 та базуються на підході REST (Representational State Transfer) на відміну від протоколу SOAP, який використовувався в Українському мовно-інформаційному фонді до цього часу для розробки застосунків на основі технології Windows Communication Foundation.

REST – це не стандарт і не специфікація, а архітектурний стиль, побудований на існуючих, добре відомих і контрольованих консорціумом W3C стандартах, таких, як HTTP, URI (Uniform Resource Identifier), XML і RDF (Resource Description Format). У REST-сервісах акцент зроблений на доступ до ресурсів, а не на виконання віддалених сервісів; у цьому їх кардинальна відмінність від SOAP-сервісів. Якщо SOAP-клієнти запитують виконання дії на сервері, то REST-клієнти просто вимагають сам ресурс.

З теоретичного погляду це начебто неістотна різниця, але на практиці вона величезна, особливо щодо інфраструктури, необхідної для підтримки кожного з цих підходів. Тут повторюється історія запозичення корпоративними intranet-мережами технологій з Internet, що відбувалася приблизно двадцять років тому. Однак є й відмінності: цілком очевидно, що архітектура WWW осмислена проєктантами застосунків набагато меншою мірою, ніж свого часу архітектура Internet. У тому трактуванні Web, яке набуло поширення і посилено експлуатується (особливо, коли мова йде про Web-сервіси), все зводиться до подання Web виключно як комунікаційного середовища. Web-сервіси, побудовані

на основі протоколу SOAP, не використовують можливості Web повною мірою – в тому числі, і можливості механізму адресації інформаційних ресурсів, заснованого на універсальних ідентифікаторах ресурсів (URI).

Не дивно, що в академічному середовищі і в близькому до нього інженерному співтоваристві, що глибше розуміє ідеї WWW, виник альтернативний напрям, який, на думку його прихильників, повноцінно, по-справжньому використовує цей чудовий винахід. Такі альтернативні сервіси називають сервісами на основі REST.

Стиль REST методично розкрито в дисертації Філдінга [2], перші чотири розділи якої присвячені визначенню поняття «архітектурний стиль» взагалі і принципам побудови мережевого програмного забезпечення.

У п'ятому розділі викладені основи REST.

Представлення отримує дані з контролера та генерує користувацький інтерфейс для відображення потрібної інформації. Для формування представлень було використано мову розмітки HTML5, яка надає користувачам широкі можливості, дозволяючи виконувати на боці клієнта завдання, що раніше виконувалися виключно сервером. Для опису зовнішнього вигляду представлень використано формальну мову CSS3. Інтерфейс, сформований цими інструментами, доступний до відображення сучасними веб-переглядачами, а у поєднанні з функціоналом мови JavaScript відрізняється зручністю, функціональністю та відповідністю сучасним веб-стандартам.

Використання таких можливостей HTML5 та CSS3 як семантичні теги розмітки, fluid верстки на основі технології Flexible Box Layout, CSS3 Multiple column layout, media Queries: resolution feature, дозволило реалізувати адаптивний інтерфейс, який пристосовується до різного розширення дисплеїв пристроїв, на яких відображується Система. Розглянемо докладніше ці інструменти.

Семантичні теги розмітки описують власний зміст. Зазначені засоби роблять більш зрозумілими та прозорими елементи інтерфейсу для пошукових систем, браузерів та самих розробників.

Flexible Box Layout Module – специфікація CSS, яка дозволяє контролювати розмір, порядок, вирівнювання елементів за двома осями, задавати правила поділу вільного простору між ними, тощо. Цей інструмент забезпечує можливість Системи адаптуватися до зміни розміру вікна. Недоліком використання Flexible Box Layout Module є відсутність підтримки в застарілих браузерах (гарантується робота Системи в браузерах Mozilla Firefox v. 30+; Google Chrome v. 30+; Opera v. 12.1+; Safari v. 7+; Internet Explorer v. 11+, Microsoft Edge).

Для того, щоб контролювати поведінку інтерфейсу під час відображення на різних пристроях, важливим інструментом є Media Queries – частина специфікації CSS3, яка дозволяє уточнити область дії css-селектора. За допомогою цього засобу є можливість явно вказати, в яких випадках застосовуються ті чи інші правила.

Вигляд інтерфейсу системи відповідно до розширення пристрою наведено нижче (рис. 1-3).

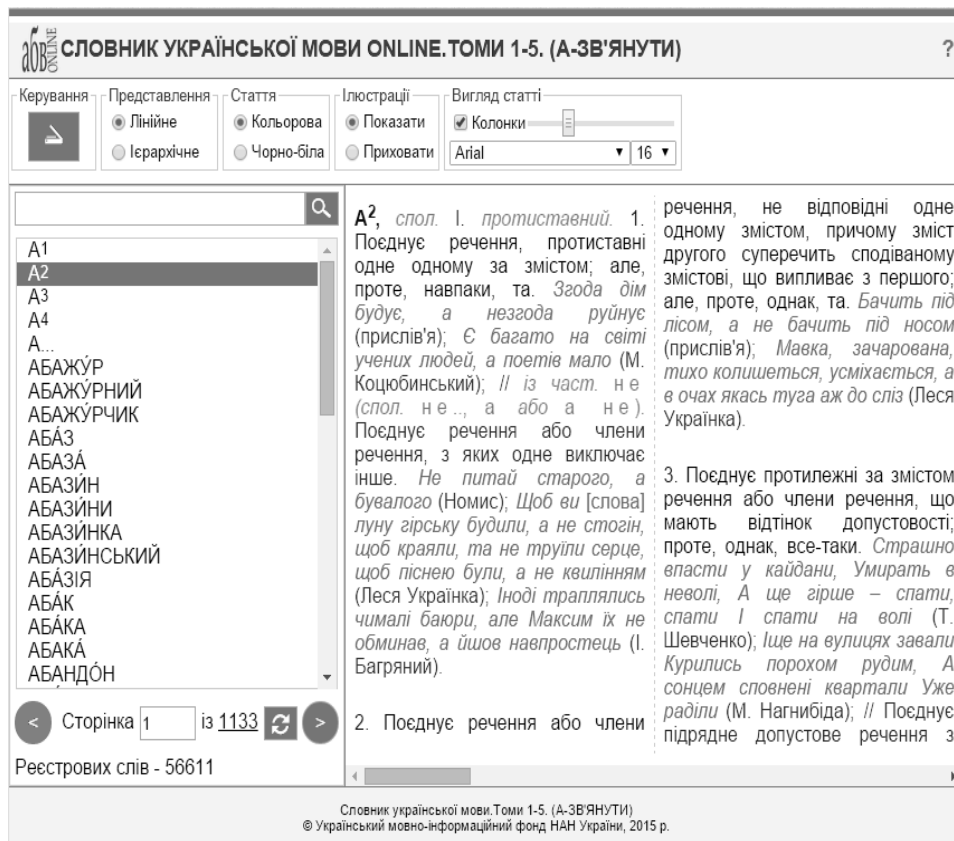


Рис. 1. Зовнішній вигляд Словника української мови on-line для пристроїв висота дисплея яких не перевищує 500 пікселів з портретною орієнтацією, та для пристроїв з альбомною орієнтацією, ширина дисплея яких не перевищує 300 пікселів

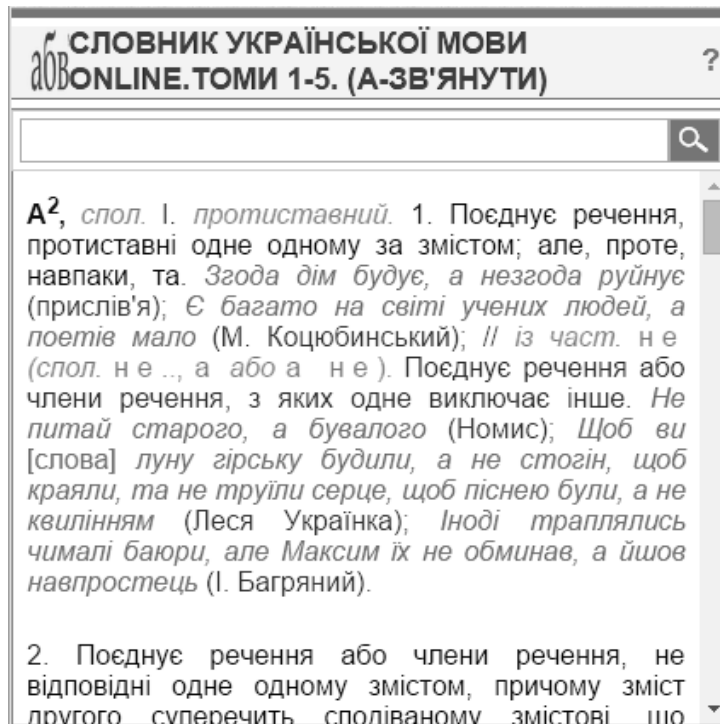


Рис. 2. Зовнішній вигляд Словника української мови on-line для пристроїв висота дисплея яких не перевищує 500 пікселів з портретною орієнтацією, та для пристроїв з альбомною орієнтацією, ширина дисплея яких не перевищує 300 пікселів

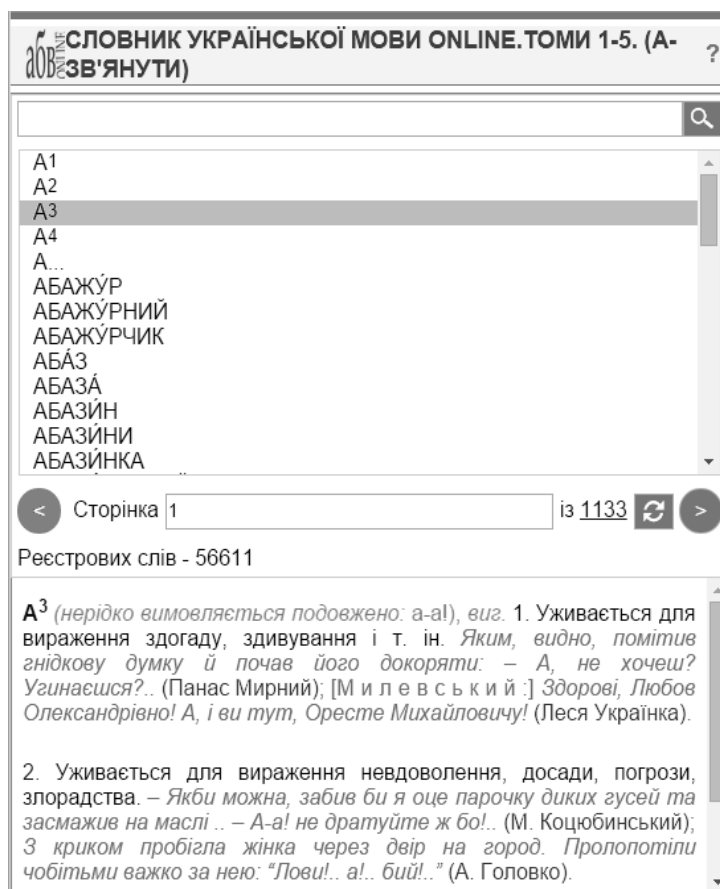


Рис. 3. Зовнішній вигляд Словника української мови on-line для пристроїв висота дисплея яких не перевищує 500 пікселів або 800 пікселів для портретної орієнтації

CSS3 Multiple column layout – багатоколонкова відображуватися в одній, двох, трьох, чотирьох, п'яти верстка контенту, завдяки якій словникова стаття може колонках (рис. 4).

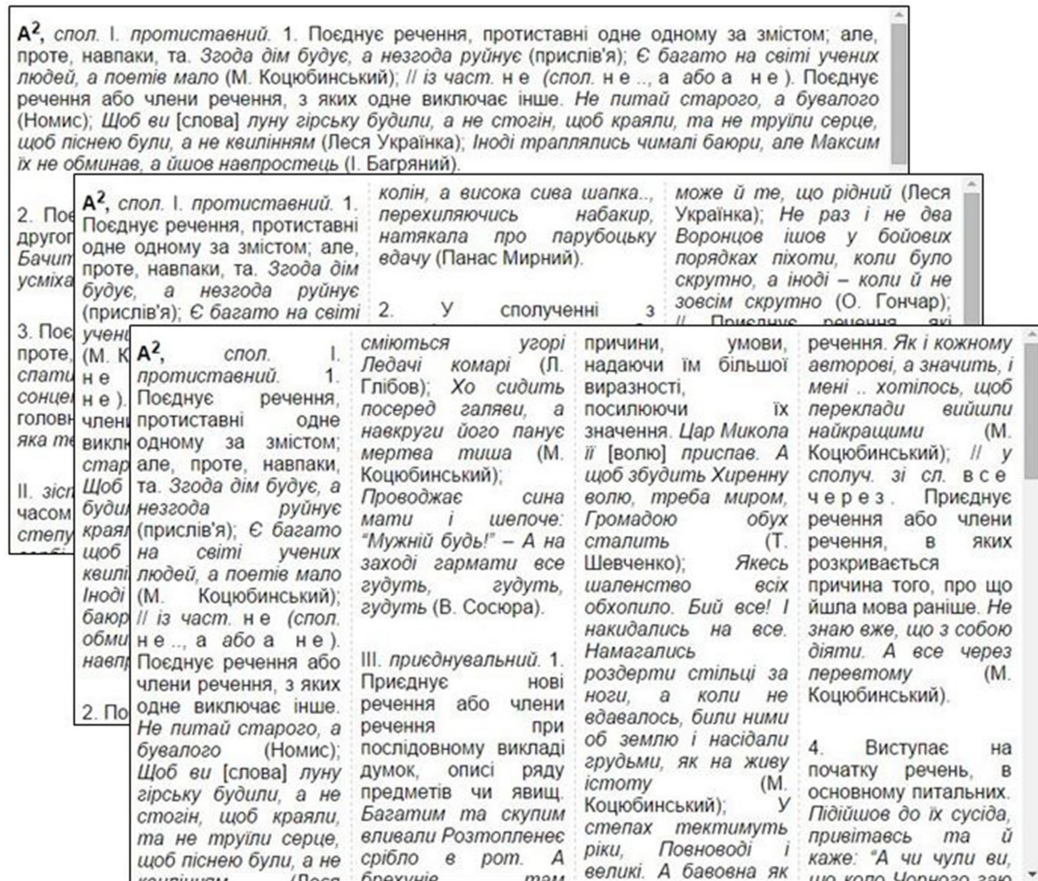


Рис. 4. Багатоколонкове представлення словникової статті Словника української мови on-line

### Висновки

Отже, технології та підходи, використані під час проектування on-line версії Словника української мови дозволили представити лексикографічний матеріал у зручному для користувача вигляді незалежно від того, який пристрій чи операційну систему він використовує; не потребують встановлення додаткового програмного забезпечення; забезпечують прийнятний рівень часу відповіді системи та можливість обслуговувати велику кількість користувачів. Враховуючи раніше наведені переваги, можна стверджувати, що використання описаного інструментарію є доцільним під час створення систем представлення лексикографічного матеріалу в мережі Інтернет.

**Список літератури:** 1. Erl, T. SOA: Principles of Service Design, Inc., 2008. – 608 pages. ISBN: 007-6092043232, ISBN-10: 0132344823. 2. Fielding, Architectural Styles and the Design of Network-based Software Architectures. Dissertation for the degree of doctor of philosophy in Information and Computer Science. 3. Fowler, M. Patterns of Enterprise Application Architecture.– 1 edition. – Boston, 2002.– 560 p. 4. Kavis, M.J. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, & IaaS). – Wiley, 2014. – 224 pages. ISBN 1118617614, 978-1118617618; 5. ASP.NET MVC 4 с прикладами на C# 5.0 для професіоналів. – М.: Издательский дом «Вільямс», 2013. – 666 с. ISBN 978-5-8459-1867-3, 978-1-4302-4236-9.

Поступила в редакцію 14.03.2016.

УДК 658.012.01156

**Системотехнічні основи створення on-line версії Словаря українського мови** / И.Н. Билиятинская // Бионика интеллекта: научн.-техн. журнал. – 2016. – № 1 (86). – С. 3-7.

При разработке систем представления лексикографического материала в электронном виде с целью более широкого охвата аудитории предлагается использовать открытые стандарты и технологии, которые обеспечивают работу пользователей независимо от характеристики их устройств (программного и аппаратного обеспечения, разрешения экрана и т.п.). Проанализированы концепции проектирования, подходы, платформы разработки, спецификации и стандарты, применяемые для разработки on-line версии Словаря украинского языка. Предложено использовать описанные технологии во время разработки систем представления лексикографической информации в сети Интернет.

Ил. 4. Библиогр.: 5 наим.

УДК 658.012.01156

**Systems Engineering of online version of the Dictionary of the Ukrainian language** / I.N. Bilyatynska // Bionics of Intelligence: Sci. Mag.. – 2016. – № 1 (86). – P. 3-7.

To enlarge the amount of the users of electronic dictionaries recommendable for the developers to apply open standards and technologies which work independently of the characteristics (hardware and software, screen resolution, etc.) of user's devices. The design concepts, approaches, development platforms, specifications and standards used in the development of on-line version of the Dictionary of the Ukrainian language were analyzed. These technologies are proposed to be used for developing the systems of lexicographical information representation in the Internet.

Fig. 4. Ref.: 5 items.