

цій технології та її перспективи. Щоб оцінити перспективи, достатньо порівняти динаміку росту обраної технології та технології, що вже завоювала прихильність за період етапу молодості. Якщо динаміка схожа, то і перспективи теж схожі, оскільки однакові темпи росту перспективної технології та тої, що вже завоювала ринок, свідчать про можливість циклічного повторення успіху.

У ході проведення дослідження виділено параметри, які не характеризують технології ні з погляду етапу життєвого циклу, ні з погляду доцільності обрання технології на різних етапах існування, але безпосередньо впливають на тенденції розвитку. Такими параметрами є: швидкість освоєння, підтримка великим компаніями.

Швидкість освоєння є дуже критичним параметром, оскільки потреба у спеціалістах з розробки програмних продуктів зростає швидше, ніж освітні установи можуть достатньою мірою випустити кваліфікованих спеціалістів. Тому простота освоєння у багатьох випадках стає тим фактором, який забезпечує ріст спільноти, появу інноваційних засобів розробки, тривалий період існування. Складність розробки зростає швидше, ніж продуктивність програмістів. Велика кількість абстракцій, які були популярні під час розробок програмних додатків останніх 30 років, призвела до того, що багато паттернів проектування є прихованими, а значить незрозумілими. Це породжує складність у розумінні молодими спеціалістами того, як той чи інший механізм працює. В умовах дефіциту професіоналів – це проблема, на яку потрібно зважати в оцінці технології. Перехід багатьох спеціалістів з розробки програмного забезпечення за допомогою C++ на Java технологію відображає цю проблему. За абсолютної переваги C++ у плані продуктивності, вимогами до пам'яті, величиною спільноти та кількістю засобів розробки Java володіє трьома перевагами: швидкість розробки, легкість освоєння, кросплатформність.

Підтримка великих компаній полягає у стрімкому зростанні на початкових стадіях, зосередження внутрішніх продуктів на використанні власної технології, що загалом її розвиває. Також істотний вплив на життєвий цикл технології відіграють кошти, які вкладуються у популяризацію. Великі компанії здійснюють підтримку мови програмування та зацікавлені у тривалому терміні її існування. Для таких мов, як Java, C#, Swift, Objective-C, підтримка батьківських компаній зіграла важливу роль в їх розвитку та позиції на ринку. Тому при схожості вихідних параметрів різних технологій підтримка компаній є істотною перевагою.

Висновки. Проаналізовано основні показники життєвого циклу технології. Встановлено закономірності притаманності певних показників відповідним етапам життєвого циклу технології. Запропоновано узагальнену класифікацію показників оцінки та перспективності використання програмних технологій. Здійснено теоретичне обґрунтування обраних показників.

Література

1. Великий тлумачний словник сучасної української мови (з дод. і допов.) / укл. і голов. ред. В.Т. Бусел. – Київ-Ірпінь : Вид-во ВТФ "Перун", 2005. – 1728 с.
2. Роганов Е.А. Основы информатики и программирования : учебн. пособ. / Е.А. Роганов. – М. : Изд-во МГИУ, 2001. – Вузол VI.

3. Popularity of Programming Language // GitHub. – 2016. [Electronic resource]. – Mode of access <http://www.pypl.github.io/PYPL.html>.
4. A small place to discover languages in GitHub // GitHub. – 2015. [Electronic resource]. – Mode of access <http://www.github.info>.
5. An Interview with the Creator of Ruby // O'Reilly Media. – 2001. [Electronic resource]. – Mode of access <http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html>.
6. Сейбел П. Кодеры за работой. Размышления о ремесле программиста / Питер Сейбел. – СПб. : Изд-во "Символ-Плюс", 2011. – 544 с.
7. TIOBE Index // TIOBE. – 2016. [Electronic resource]. – Mode of access http://www.tiobe.com/tiobe_index.
8. Креншоу Д. Давайте создадим компилятор! / Джек Креншоу. – М. : Изд-во "Наш мир", 2001. – 465 с.

Надійшла до редакції 16.05.2016 р.

Пасека Н.С., Турчин О.Б. Теоретическое обоснование показателей жизненного цикла технологий

Проанализированы основные показатели жизненного цикла технологий и выделена обобщенная система для оценки этапа жизненного цикла. Показатели рассмотрены как свойства, которыми должна обладать та или иная технология, чтобы соответствовать определенному этапу своего существования. Исследованы существующие подходы анализа развития технологии. Предложена обобщенная классификация, что представляет собой перечень показателей для соответствующего этапа развития. Обобщенная система оценки технологий позволит минимизировать технологические риски, связанные с кратко- и долгосрочными перспективами разработки программного обеспечения.

Ключевые слова: жизненный цикл, технология, параметры, этап развития, обобщенная классификация, технологические риски, язык программирования.

Paseka M.S., Turchyn O.B. Theoretical Explanation of Technologies Life Cycle Indicators

The basic indicators of the life cycle of technology were analyzed and the generalized system for the evaluation of the life cycle phase is allocated. Indicators are considered as properties that technology must have to fit a certain stage of its existence. The existing approaches of technology development analysis are studied. The generalized classification, which is a list of indexes for the proper stage of development, is proposed. The generalized technologies estimation system will enable to minimize technological risks related to the short-term and long-term software development prospects.

Keywords: life cycle, technology, stage of development, generic classification, technological risks, programming language.

УДК 004.9:530.1

МЕТОДИ ТА ЗАСОБИ ОПРАЦЮВАННЯ ВЕЛИКИХ ДАНИХ У СИСТЕМАХ ТЕРИТОРІАЛЬНОГО УПРАВЛІННЯ

Ю.Я. Болубаш¹

Подано визначення Великих даних та описано основні характеристики. Проаналізовано математичні засоби подання та опрацювання Великих даних та визначено їх обмеження. Визначено формальний опис Великих даних. Подано моделі асоціацій між сутностями та характеристиками для різних категорій NoSQL баз даних. Розроблено метод обміну різнотипними даними та приведення реляційних даних до моделі "сутність-характеристика". Спроектовано схему даних регіону. Апробовано розроблені методи та алгоритми.

¹ Здобувач Ю.Я. Болубаш – НУ "Львівська політехніка"

Ключові слова: Великі дані, інформаційна модель, гетерогенні дані.

Вступ. Регіон – складна, багатокomпонентна, відкрита, динамічна, ймовірнісна система, оскільки:

- до складу регіону входять різні за природою об'єкти, такі як економічні показники, соціальні фактори, технічні об'єкти, які, водночас також є системою;
- стан системи змінюється під дією зовнішніх факторів;
- перехід з одного стану в інший відбувається не миттєво, а з часом;
- причинно-наслідкові зв'язки є ймовірнісного характеру.

Характеристики регіону (рис. 1):

- великий набір сутностей: особи, організації, природні ресурси, рекреаційний фонд, законодавчі акти та звіти;
- база даних особливостей: документи для інтелектуального аналізу даних, словники даних для зв'язування об'єктів.

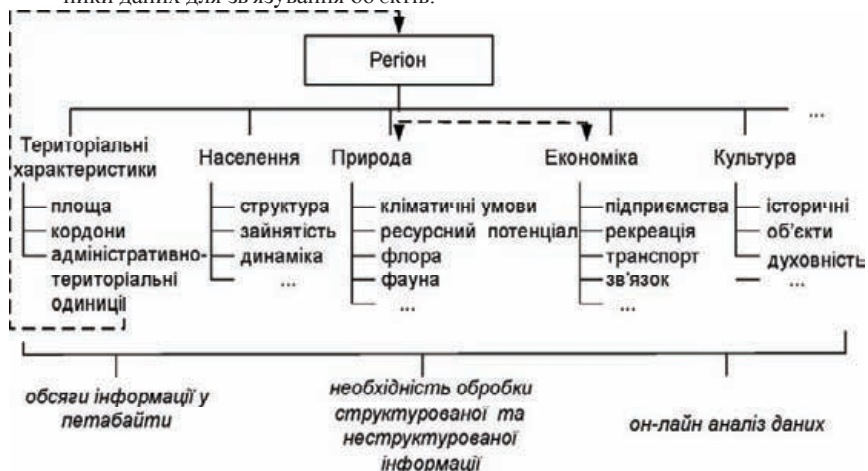


Рис. 1. Структура регіону

Процес побудови узагальненої (комплексної) моделі регіону ускладнюється різноманітністю моделей даних, а також через наявність різних рівнів агрегації даних. Однією з популярних технологій для розроблення систем територіального управління є Великі дані.

Аналіз літературних джерел та постановка задачі. Великі дані (Big Data) в інформаційних технологіях, за визначенням К. Лінч, Д. Ленеї, – набір методів та засобів опрацювання структурованих і неструктурованих різнотипових динамічних даних великих обсягів з метою їх аналізу та використання для підтримки прийняття рішень [1]. Є альтернативою традиційним системам управління базами даних і рішенням класу Business Intelligence. До цього класу відносять засоби паралельного опрацювання даних (NoSQL, алгоритми MapReduce, Hadoop) [2].

Визначальними характеристиками для Великих даних є [2]: обсяг (volume, у сенсі величини фізичного обсягу), швидкість (velocity у сенсах як швидкості приросту, так і необхідності високошвидкісного оброблення та отримання

результатів), різноманіття (variety, у сенсі можливості одночасного оброблення різних типів структурованих і слабоструктурованих даних).

Великі дані є терміном, який використовують для ідентифікації наборів даних, з якими не можна впоратися з використанням існуючих методологій та програмних засобів через їх великий розмір і складність. Такі дослідники М. Гілберт [3], С. Стрінваса та ін. розробили методики і програмні засоби для передачі даних або видобування інформаційних гранул з Великих Даних (колекції об'єктів, які зазвичай формуються для атрибутів з числовими і які розташовані поряд через їх схожість, функціональну або фізичну суміжність). Методи машинного навчання та візуалізації даних дають змогу опрацювати та графічно подати результати аналізу даних великих обсягів (мільйони кортежів). Проте нерозв'язаною задачею залишається задача побудови відображення між моделями даних різних джерел.

У роботах Alejandro Maté [4] та Lucentia Research Group обґрунтовано використання багатовимірної моделі для подання Великих даних та побудови відображення в реляційну модель. Багатовимірне подання даних добре використовувати для задач візуалізації даних та їх аналізу, але у зв'язку з розрідженістю гіперкуба обсяг даних у такому випадку є більший порівняно з реляційним поданням, що є неприпустимим до Великих даних.

Vinayak Borkar, Yingyi Bu [5] пропонують використовувати об'єктно-орієнтований підхід, проте обмеженням є кількість зв'язків між об'єктами. Тому об'єктне подання даних за певної модифікації може бути використане для Великих даних. Проте залишається нерозв'язаною задача трансформації з одних типів подання даних в об'єктну модель даних (табл. 1).

Табл. 1. Порівняння моделей подання Великих даних

Назва моделі	Автори	Переваги	Недоліки
Багатовимірна модель	М. Слокун	Доцільно використовувати для задач візуалізації даних та їх аналізу	У зв'язку з розрідженістю гіперкуба з неоднорідними даними, їх обсяг збільшується, що є неприпустимим до Великих даних
Об'єктна модель	І. Добеші	За певної модифікації може бути використана для Великих даних	Нерозв'язаною є задача трансформації з одних типів подання даних в об'єктну модель даних
Графова модель	Чен [6], Паркер, Демченко	Доцільна для аналізу зв'язків між об'єктами	Значна обчислювальна складність алгоритмів пошуку за умови великої кількості об'єктів

Отже, єдиного підходу до опрацювання Великих даних не знайдено. Тому задача розроблення методів та засобів опрацювання Великих даних у системах регіонального розвитку є актуальною.

Основний матеріал. Отже, інформаційна модель Великих даних – це

$$BigD = \langle e, f, a \rangle,$$

де сутності $e \in E$, характеристики $f \in F$, асоціації $a \rightarrow n_{e,f}$ між сутностями e та характеристиками f .

Загальна кількість сутностей визначається як $|E|$, загальна кількість характеристик є потужністю множини $F: |F|$. Також опишемо:

- для кожної характеристики f множини $e(f) = \{e \in E : n_{e,f} > 0\}$ усіх асоційованих з f сутностей;
- для кожної сутності e множини $f(e) = \{f \in F : n_{e,f} > 0\}$ усіх асоційованих з e характеристик.

Опишемо ці якісні подання у кількісному вигляді [8, 9]. Для цього скористаємося аналогом опису міри TF-IDF у текстових документах [10].

У подібних ситуаціях, коли є кілька сутностей, пов'язаних з характеристикою, використаємо кількісне подання інформації, тобто кількість бінарних запитань (так, ні), які потрібно задати, щоб знайти потрібний об'єкт. Загалом, якщо знаємо, що невідомий об'єкт належить множині, що складається з N елементів, то можемо розділити цей набір на дві половини і, задаючи двійкові питання, з'ясувати, до якої половини належить шуканий об'єкт. Отже, тоді кількість об'єктів становитиме $N/2$. Продовжимо далі таку ж процедуру: задамо друге питання, для чого поділимо виділену половину ще на дві половини. Отже, після двох запитань матимемо $N/4$ об'єктів, серед яких є шуканий. Після трьох запитань матимемо $N/8$. Загалом, після відповіді на q бінарних запитань матимемо множини з $N \cdot 2^{-q}$ елементів, що містить потрібний об'єкт [11].

Коли множина складатиметься з одного елемента, точно визначимо потрібну нам альтернативу. Кількість бінарних запитань для пошуку характеристики для N альтернатив: $N \cdot 2^{-q} = 1$, або $q = \log_2(N)$.

Аналогічно можна описати сутності. Маємо $|E|$ сутностей з кількістю інформації $\log_2(|E|)$. Коли знаємо, що якась сутність асоційована з характеристикою (маємо $|e(f)|$ сутностей асоційованих з характеристикою f), то кількість питань дорівнює $\log_2(|e(f)|)$. Отже, той факт, що сутність e пов'язана з характеристикою f , дає змогу зменшити кількість питань до

$$k = \log_2(|E|) - \log_2(|e(f)|) = \log_2\left(\frac{|E|}{|e(f)|}\right).$$

Загальна важливість характеристики f для сутності e визначається як $\log_2\left(\frac{|E|}{|e(f)|}\right)$ з фактором важливості $1 + \log_2(n_{e,f})$. Результируюча кількість питань визначається як

$$I(e, f) = (1 + \log_2(n_{e,f})) \cdot \log_2\left(\frac{|E|}{|e(f)|}\right).$$

Ця формула є одним з варіантів у термінах частоти термінів – т. зв. зворотною частотою документа tf-idf [11]. Для кожної сутності e маємо кількість питань $I(e, f)$ для різних характеристик f . Значення важливості потрібно нормалізувати:

$$V(e, f) = \frac{(1 + \log_2(n_{e,f})) \cdot \log_2\left(\frac{|E|}{|e(f)|}\right)}{\sqrt{\sum_{j \in f(e)} \left((1 + \log_2(n_{e,j})) \cdot \log_2\left(\frac{|E|}{|e(j)|}\right) \right)^2}}.$$

Для кожної сутності e є вага $V(e, f)$. Отже, в якості міри близькості між двома об'єктами E_1 і E_2 , можемо вважати відстань між відповідними векторами $(V(e_1, f), V(e_2, f), \dots)$.

У звичайній Евклідовій відстані $d(a, b) = \sqrt{(a_1 - b_1)^2 + \dots}$ додаються квадрати різниць. Отже, для кожної ваги $V(e, f)$, що репрезентує кількість відповідей "так"- "ні", матимемо

$$d(e_1, e_2) = \sum_{f \in F} |V(e_1, f) - V(e_2, f)|.$$

Ця відстань залежить від кількості характеристик: наприклад, якщо на додаток до документів, зберігаємо їх копії, відстань збільшується вдвічі. Щоб уникнути цієї залежності, відстань $d(e_1, e_2)$, як правило, нормалізується в інтервалі $[0, 1]$ через ділення на максимально можливе значення цієї відстані.

Порівнюючи сутності по кожній з характеристик, отримуємо відстань між ними, подану як

$$D(e_1, e_2) = \frac{\sum_f |V(e_1, f) - V(e_2, f)|}{\sum_f \max(V(e_1, f), V(e_2, f))}.$$

Далі здійсимо подання відомих моделей даних у моделі даних "сутність-характеристика". Носій даних у моделі "ключ-значення" (інша назва – колонкова БД) описується кортежами виду

$$KV = \{ \langle k, v \rangle \},$$

де: k – ключ, який приймає унікальні значення у кожній парі; v – значення, що відповідає цьому ключу, $e \leftrightarrow k; f \leftrightarrow v$. Сигнатура моделі виглядає як:

$$O = \langle \pi, \sigma \rangle,$$

де: π – операція проєкції за атрибутами (ключ або значення); σ – селекції атрибутів (вибір значення за ключем, ключів за значенням, ключів за значенням предків). Перераховані операції відносять до категорії читання [9].

Приклади реальних операцій читання:

- get(key),
- multiGet,
 - також MultiGetIterator, StoreIterator (за major key)
 - Subrange (keyFirst, keyLast)
 - Depth.CHILDREN_ONLY
- multiGetKeys
 - Subrange (keyFirst, keyLast)
 - Depth.CHILDREN_ONLY

Прикладом СУБД колонкового типу є Cassandra.

Носій моделі "об'єкт-документ" описується кортежами виду

$$OD = \{ \langle f_0, \langle f_1 : e_1, f_2 : e_2, f_n : e_n, f_{n+1} : d_1, f_2 : d_2, f_{n+1} : d_l \rangle \},$$

де: f_0 – ідентифікатор документа; $f_1..f_m$ – характеристики (атрибути) документу; $e_1..e_m$ – атомарні значення характеристик $f_1..f_m$; $d_1..d_l$ – посилання на інші документи, $d_i = e(f_i)$. Операції цієї моделі є об'єктні.

Операція визначення вузлів елемента

$$v(f_i) = \{C\} \cup \{f_0, | i = \overline{1, n}\} \cup \{e(f_i) | i = \overline{0, n+l}\},$$

де C – колекція документів f_0 . Операція визначення значень вузлів

$$v(f_i) = \{n_{e_j, f_i} | i = \overline{1, n}, j = \overline{0, m+l}\},$$

де e_j – значення атрибутів f_i . Прикладами СУБД цього типу є MongoDB та CouchDB. Графова модель даних подана як:

$$O = \langle ID, A, z, r \rangle,$$

де: ID – множина ідентифікаторів, вузлів графа; A – множина позначених спрямованих дуг (p, l, c) , $p, c \in ID$, l – "рядок-мітка", запис (p, l, c) означає, що між вузлами p та c зв'язок l ; z – функція, що відображає кожний вузол $n \in ID$ в конкретне значення складеного або атомарного типу, $z : n \rightarrow v$; V – особливий кореневий вузол графа.

Основною метою перетворення даних у модель "сутність-характеристика" є забезпечення можливості опрацювання даних будь-якої структури.

Метод перетворення даних у модель "сутність-характеристика" полягає у перерахунку важливості характеристики для сутності, а також фізичному перетворенні схеми даних у пару "сутність-характеристика". Схему перетворення даних у модель "сутність-характеристика" подано на рис. 2.

Для розширення XML пропонується використовувати архітектуру *MVVM* ("View-Model", "Вигляд-Модель"). Цей шаблон застосовується під час проектування архітектури програмного додатка й використовує поділ моделі в частині логіки її функціонування та її подання на користувацькому інтерфейсі. Основною її відмінністю від відомої архітектури програмного додатка *MCV* (Model-View-Controller, Модель-Вигляд-Контролер) є відсутність вимоги прив'язування даних до їхнього подання. Відповідно до цього підходу покажемо потрібні розроблені компоненти для автоматизованої конвертації з реляційної БД у модель XML з урахуванням особливостей моделі "сутність-характеристика" (рис. 3).

В архітектурі додатків, які реалізуються за структурою, яку показано на рис. 3, присутні кілька компонентів, які потребують додаткових пояснень. Один з таких компонентів – XML-розмітка. Вказана розмітка також може бути використана для роботи з документо-орієнтованими базами даних. Тому перетворення з моделі "об'єкт-документ" у модель "сутність-характеристика" не виконалось.

Наведемо приклад XML опису вимірів системи туристичного комплексу.

```
<measurement>
<name> Станція </name>
<location> Красне </location>
```

```
<time> 2014-04-14 09:41 </time>
<brake>
<fuzzy name="brakel">
<function minvalue="0"> 0 </function>
<function minvalue="1" maxvalue="5"> x/(maxvalue-Minvalue) + maxvalue/(maxvalue-minvalue) </function>
</fuzzy>
<fuzzy name="brake2">
<function minvalue="6"> 6 </function>
<function minvalue="7" maxvalue="9"> x/(maxvalue-minvalue) - minvalue/(maxvalue-minvalue) </function>
</fuzzy>
</brake>
</measurement>
```

Для зручності універсального опису елементів можна застосовувати Xsd опис, вид якого подано нижче. Функція приналежності в цьому прикладі обмежена значеннями Minvalue і Maxvalue.

```
<?xml:namespace prefix="s" schema="http://www.w3.org/2001/XMLSchema" type="qualified" />
<xs:schema id="Fuzzyschema" elementFormDefault="qualified" />
<xs:complexType name="fuzzy">
<xs:sequence>
<xs:element name="fuzzy" minOccurs="1" maxOccurs="unbounded" type="functions7" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="functions">
<xs:sequence>
<xs:element ref="function" minOccurs="1" maxOccurs="unbounded" />
</xs:sequence>
<xs:attribute name="name" type="xs:string" />
</xs:complexType>
<xs:element name="function">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string" />
<xs:attribute name="minvalue" type="xs:string" />
<xs:attribute name="maxvalue" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</ts:element>
</xs:schema>
```

Далі розроблено метод перетворення реляційних даних у модель "сутність-характеристика" transformRDB, який містить такі кроки.

Крок 1. Створити кореневий елемент схеми RDB для моделі даних *Big Data*.

Крок 2. Для кожної сутності e створити окремий елемент Entity і розмістити його під кореневим елементом.

Крок 3. Для кожної характеристики f сутності e з важливістю, більшою за порогове значення, створити атрибут Feature, розмістити його усередині відповідного опису сутності та задати його тип.

Крок 4. В атрибуті потрібно вибрати й задати тип даних та визначити граничні значення.

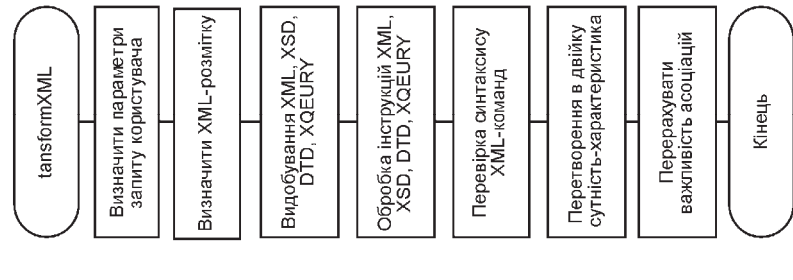


Рис. 3. Схема конвертації XML-БД з урахуванням моделі "сутність-характеристика"

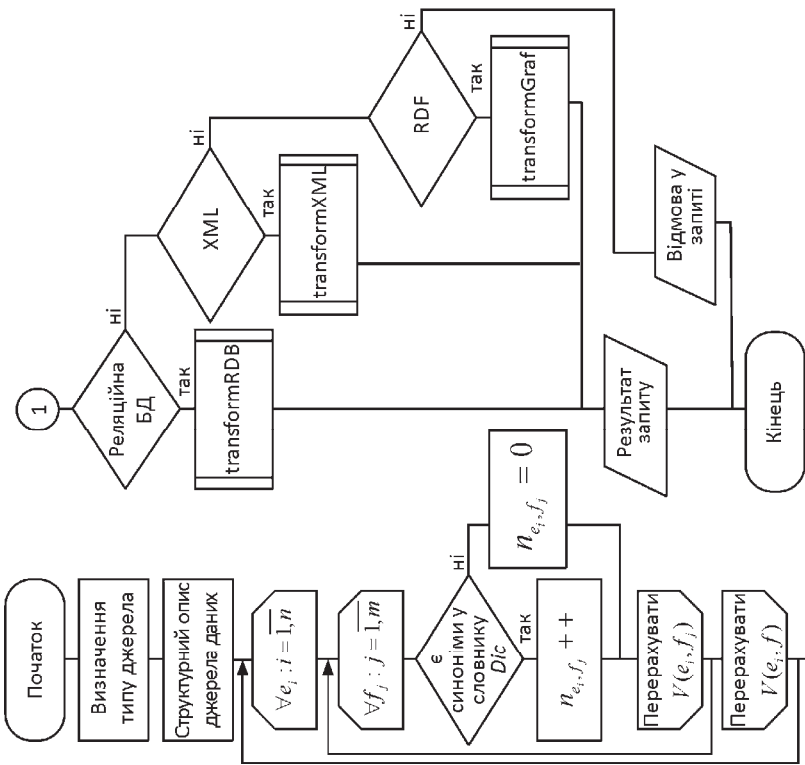


Рис. 2. Схема перетворення даних у модель «сутність-характеристика»

Метод перетворення з формату JSON нагадує структуру методу перетворення з формату XML і тому окремо не розглядаємо. Для перетворення графової моделі в модель "сутність-характеристика" важливим є визначення ваги зв'язку між елементами. Оскільки першим параметром моделі є характеристика, другим – зв'язок, а третім – сутність, то перетворення між моделями полягатиме у числовому вираженні асоціації між елементами RDF-моделі (рис. 4). Далі спроектовано архітектуру системи опрацювання даних регіону з використанням моделі "сутність-характеристика". Діаграма класів описує подання даних у моделі "сутність-характеристика" (рис. 5).

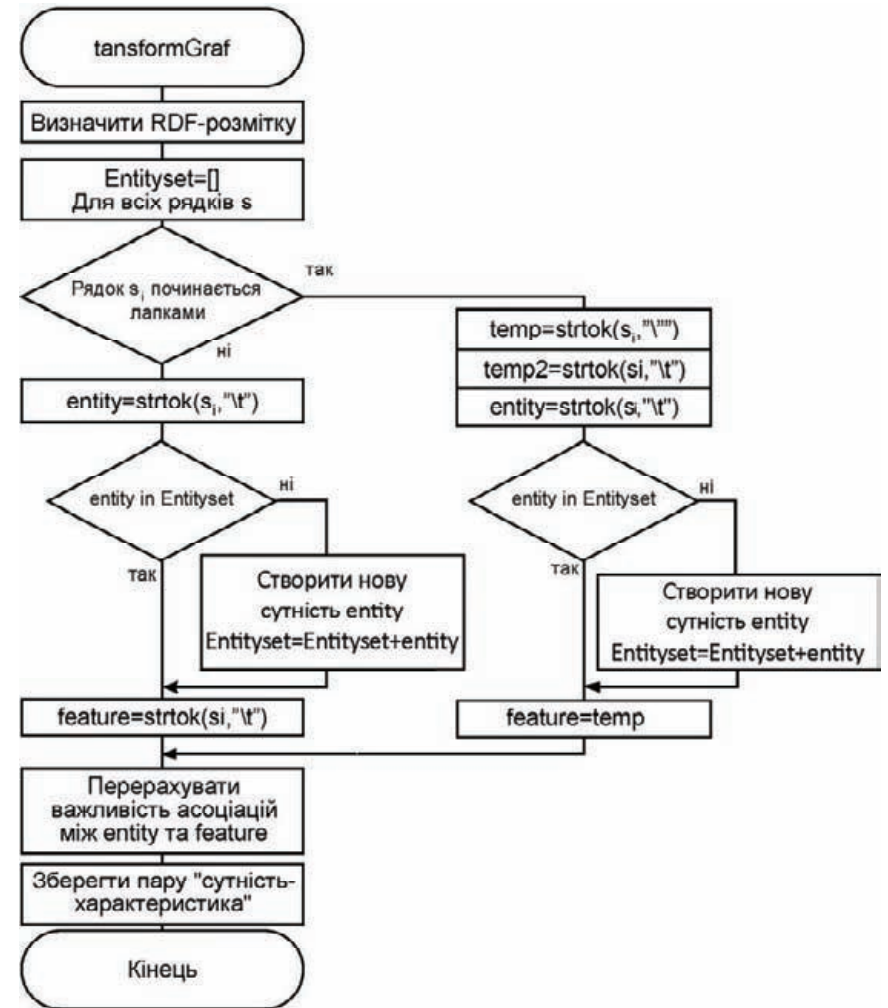


Рис. 4. Схема конвертації графової БД з урахуванням моделі "сутність-характеристика"

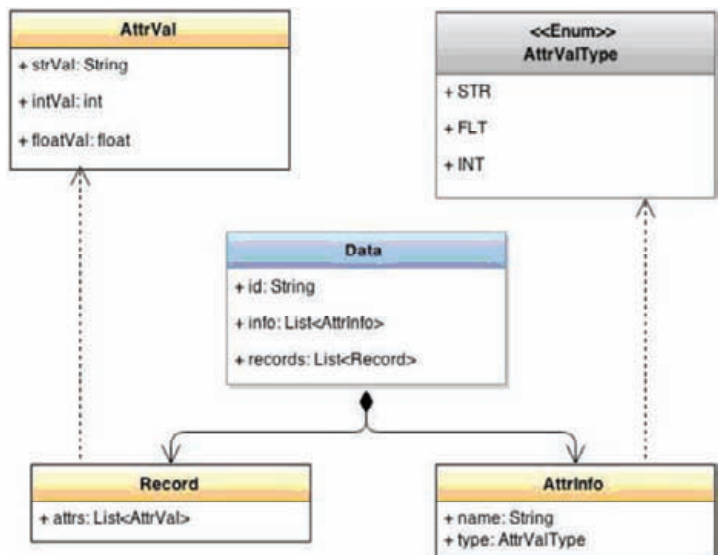


Рис. 5. Діаграма класів

Для уніфіковано перетворення запитів розроблено діаграму класів (рис. 6) та інтерфейс взаємодії (рис. 7).

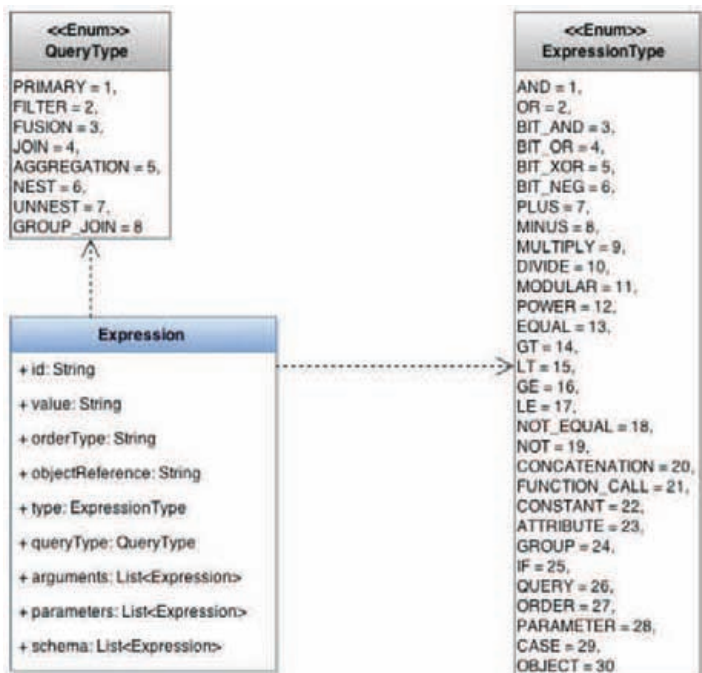


Рис. 6. Діаграма для уніфіковано перетворення запитів

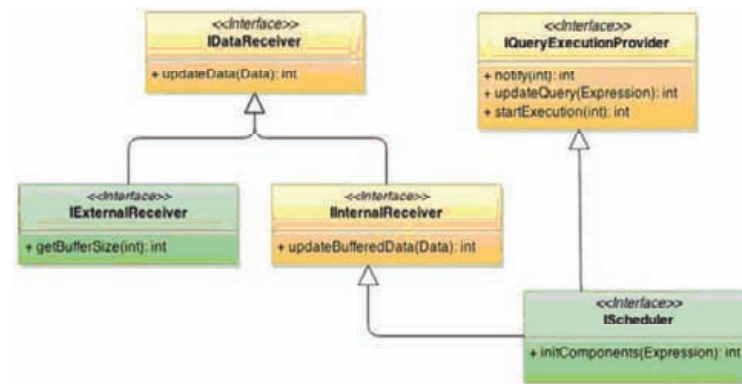


Рис. 7. Інтерфейс для уніфіковано перетворення запитів

Для визначення методики аналізу фінансових показників регіону розроблено відповідне програмне забезпечення. Для реалізації сховища даних була вибрана 64-розрядна платформа (x64). Узагальнене сховище даних складається з таких компонент:

- Реляційна база даних (Microsoft SQL Server Database Services, Oracle Database, MySQL, PostgreSQL) у 64-розрядному виконанні;
- Багатовимірні бази даних (Microsoft SQL Server Analysis Services або Hyperion Essbase) у 64-розрядному виконанні;
- "Ієрархічна" база даних (MongoDB);
- Система керування федеративним сховищем даних, яка є окремою програмою, розробленою спеціально для забезпечення функціонування сховища даних, та включає в себе файлове сховище.

Для розроблення системи керування федеративним сховищем даних було використано платформу Microsoft.Net, мову C# та середовище розробки Visual Studio. Бібліотека класів, що поставляється з .net та мова високого рівня C#, а також методологія RAD (швидкої розробки застосувань), на якій побудоване середовище розробки Visual Studio, дає змогу швидко створювати застосування, орієнтовані на бази даних.

Насамперед здійснюється федеративне опрацювання даних з джерел. Аналізуємо відносну кількість об'єктів або документів, наявних у джерелах даних, до загальної кількості об'єктів, які потрапили у федеративне сховище. У табл. 2 наведено структуру детермінованих схем БД з деталізацією за областями в обсязі, достатньому для прогнозування процесів розвитку регіону. У табл. 2 також вказано структурованість даних і розміщення у сховищі даних (порядковий номер області та тип БД).

Табл. 2. Структура детермінованих схем БД

Джерело/область/таблиця	Відстань $V(e, f)$	Розміщення
Джерело даних MFUVUDB	0.154224	-
MFUVUDB.t aspnet Applications	0,0	RDB
MFUVUDB.t mfuvu Settlements	1.0	RDB
MFUVUDB.t vw mfuvu list sti correspond	0.666942	RDB, XML

MFUVUDB.t vw mfuvu Users	0.401167	RDB, XML
Джерело даних TransBudgDB	0.628414	-
TransBudgDB.t VXXDDMMYY	0.961365	RDB
TransBudgDB.t D BUDG LOCAL DET	0.888913	RDB
TransBudgDB.t D ECON CRED	0.9	RDB, XML
TransBudgDB.t D FIN	0.916667	RDB, XML
TransBudgDB.t DMYYMM	0.533333	RDB, XML
TransBudgDB.t D INC DET	0.903839	RDB
TransBudgDB.t district	1,0	RDB
TransBudgDB.t VW INCOME LVL3	0.5	RDB, XML
Область даних FUPortalDB	0.325218	-
FUPortalDB.t Applications	0.666667	RDB, XML
FUPortalDB.t IsAuthCorrespond	1,0	RDB
FUPortalDB.t Articles	0.375858	RDB, XML
FUPortalDB.t People	0.227109	RDB, XML
FUPortalDB.t aspnet Applications	0.666667	RDB, XML

Проаналізовано повноту потрапляння інформації у федералізоване сховище даних (рис. 8).

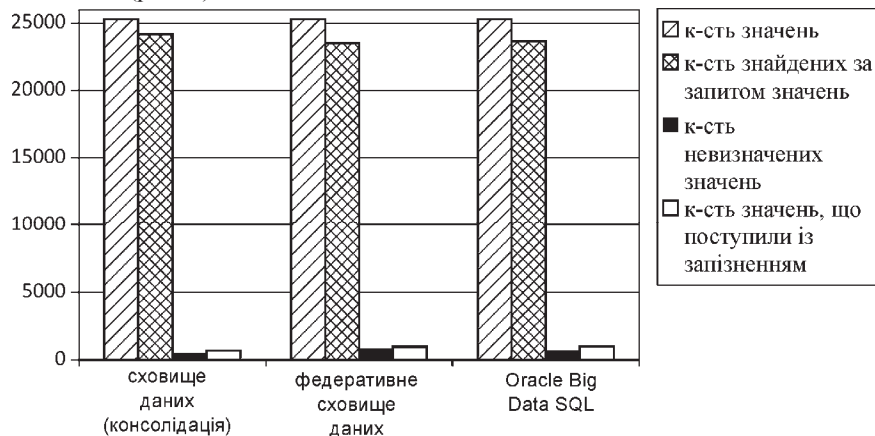


Рис. 8. Аналіз повноти нагромаджених описань об'єктів

На поданій діаграмі проаналізовано роботу алгоритму федеративного запиту. Роботу алгоритму порівняно з роботою алгоритму немодифікованої інтеграції, застосованому в Oracle Data Integrator. Дані у федеративне сховище потрапляють з баз даних різних установ, структури даних яких наперед невідомі. Кількість записів вхідних баз даних, що мають потрапити у федеративне сховище даних, – 15000.

Для аналізу повноти нагромаджених об'єктів порівнювалися традиційне сховище (використано Integrated services), Oracle Big data SQL та запропоноване рішення. Визначено, що традиційна консолідація працює найкраще, що є зрозумілим, але цей підхід неприйнятний для великих даних. Тому далі порівнювалися запропоноване рішення та Oracle Big data SQL. Визначено, що незначно домінується за кількістю невизначених значень, але домінує за кількістю

вірно знайдених даних за запитом, а також дає змогу знайти потрібні дані за короткий час (дані, що надійшли зі запізнення).

Далі проаналізовано правильність перетворення запитів різних типів. З цією метою порівнювалися розроблені засоби в системі Інтегратор з середовищем DocumentDB, у якому є можливість формування запитів на мові SQL та NoSQL. Результати порівняння подано в табл. 3 у відсотковому значенні правильно поданих запитів. Загальна кількість запитів, що тестувались – по 50 запитів кожної категорії. Правильність формування запитів перевірялася експертно.

Табл. 3. Перетворення запитів різних типів

Вид запиту	Інтегратор				DocumentDB			
	select	select... join	insert	delete	select	select... join	insert	delete
РБД (Microsoft SQL Server Database Services)	98	95	93	93	98	92	94	94
XML (XBase)	86	82	-	-	-	-	-	-
NoSQL (mongoDB)	91	86	84	84	89	82	81	81

Далі для апробації та тестування розроблених методів визначили важливість об'єктів у різних джерелах даних. Під важливістю розуміли релевантність запиту. Як і варто було очікувати, для реляційних джерел вона вища (рис. 9).

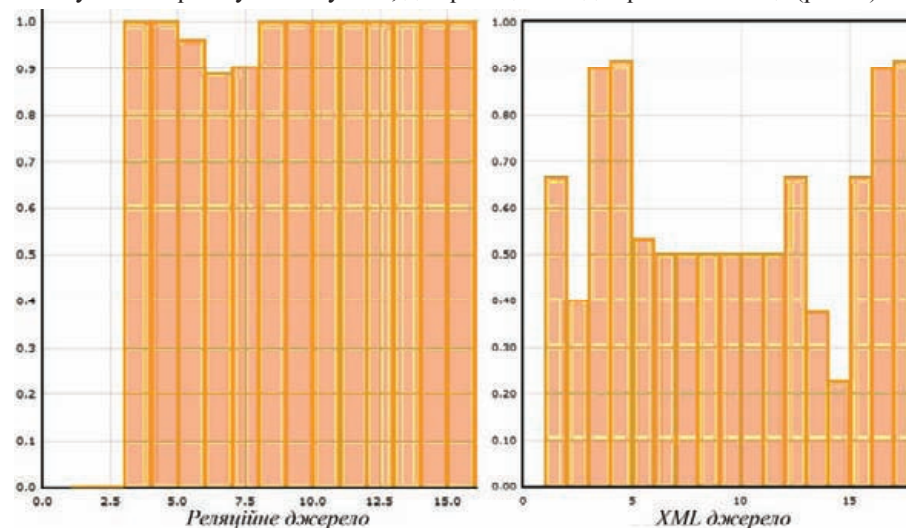


Рис. 9. Важливість сутностей

Висновки

Здійснено аналіз проблеми подання та опрацювання різномісних джерел даних. Обґрунтовано актуальність розв'язання цієї проблеми на основі введення Великих даних, що дало змогу виділити невирішені раніше задачі з опрацювання та консолідації даних з наперед невідомих джерел. Розроблено модель

Великих даних "сутність-характеристика", яка дає змогу організувати структуровані та слабоструктуровані дані і на відміну від багатовимірної моделі не містить надлишковості.

Література

1. Laney, D. (2012). *The Importance of 'Big Data': A Definition*. Gartner, 242 p.
2. Beyer, M. (2011). *Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data*. Gartner. Archived from the original on 10 July 2011.
3. Hilbert, Martin. (2013). *Big Data for Development: From Information – to Knowledge Societies*, SSRN Scholarly Paper No. ID 2205145). Rochester, NY: Social Science Research Network; <http://papers.ssrn.com/abstract=2205145>.
4. Maté, A., Peral, J., Ferrández, A., Gil, D., & Trujillo, J. (2016). A hybrid integrated architecture for energy consumption prediction. *Future Generation Computer Systems*.
5. Borkar, Vinayak, & Michael, J. (2012). Carey, and Chen Li. "Inside Big Data management: ogres, onions, or parfaits?." Proceedings of the 15th international conference on extending database technology. ACM.
6. Bu, Yingyi, et al. (2010). HaLoop: efficient iterative data processing on large clusters. *Proceedings of the VLDB Endowment 3.1-2*, 285-296 pp.
7. Min, Chen, Mao, Shiwen & Liu, Yunhao. (2014). Big data: a survey. *Mobile Networks and Applications 19.2*, 171-209 pp.
8. Shakhovska, N., & Bolubash, Yu. (2015). Big Data Model "entity and characteristics". *Econ-TechMod*, Vol. 4, No. 2, 51-58 pp.
9. Шаховська, Н.Б. & Болюбаш, Ю.Я. (2015). Модель Великих даних "сутність-характеристика". *Вісник Національного університету "Львівська політехніка". – Сер.: Інформаційні системи та мережі, 814*, 186-196 сс.
10. Di Ciaccio, A., Coli, M., & Angulo Ibanez, J.M. (eds.) (2012). *Advanced Statistical Methods for the Analysis of Large Data*. Springer, Berlin
11. Fang, L., Sarma, A.D., Yu, C., & Bohannon, P. (2011). Rex: explaining relationships between entity pairs. *Proc. VLDB Endowment, 5(3)*, 241-252 pp.

Надійшла до редакції 26.06.2016 р.

Болюбаш Ю.Я. Методы и средства обработки Больших данных в системах территориального управления

Дано определение Больших данных и описаны их основные характеристики. Проанализированы математические средства подания и обработки Больших данных и определены их ограничения. Определено формальное описание Больших данных. Показаны модели ассоциаций между сущностями и характеристиками для различных категорий Nosql баз данных. Разработан метод обмена разнотипными данными и приведение реляционных данных к модели "сущность-характеристика". Спроектирована схема данных региона. Апробированы разработанные методы и алгоритмы.

Ключевые слова: Большие данные, информационная модель, гетерогенные данные.

Bohubash Yu. Ya. The methods and tools for Big data processing in territorial administration systems

The paper presents Big data information model and describes the main characteristics. Mathematical analysis means submission and processing of large data are given and their limitations defined. The formal description of Big Data is built. There is posted models associations between entities and properties for the different categories Nosql databases. The method of heterogeneous data sharing and bringing to relational data model "entity-characterization" is given. The data scheme of region is designed. The developed methods and algorithms are tested.

Keywords: big data, information model, heterogeneous data.

УДК 004.942:537.86/87

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ РОЗПОДІЛУ ЕЛЕКТРОМАГНІТНОГО ПОЛЯ НАД ОДНОВИМІРНОЮ ПЕРІОДИЧНОЮ СТРУКТУРОЮ

І.В. Ничай¹

Розглянуто блок-схему алгоритму розрахунку просторового розподілу електромагнітного поля, збудженого стороннім джерелом у присутності одновимірної періодичної структури. Запропоновано реалізацію алгоритму у середовищі Matlab. Наведено приклади проведених розрахунків залежності напруженості поля від просторового кута для двох величин періоду структури. Показано можливість використання розробленої комп'ютерної моделі для дослідження особливостей формування поля при зміні конструктивних параметрів діелектричної пластини, діелектрична проникливість якої модульована однією періодичною послідовністю прямокутних функцій, та частоти збудження джерела електромагнітного поля.

Ключові слова: одновимірна періодична структура, діелектрична пластинка, електромагнітне поле, математична модель, комп'ютерне моделювання.

Вступ. Одну з перших ґрунтовних робіт, в яких досліджено поширення електромагнітних хвиль періодичною структурою, видано в середині ХХ ст. [1]. З моменту появи цієї праці минуло чимало часу, однак така тематика не втрачає своєї актуальності. Цей факт зумовлений появою нових технологій, що значно розширюють інструментарій досліджень, можливості втілення теорії на практиці. Порівняно з фізичним експериментом, який потребує створення макету досліджуваного об'єкта, математичне моделювання є більш економічно вигідним, однак вимагає наявності моделей, які б адекватно описували ту властивість об'єкта, яка цікавить дослідника. Треба зазначити, що такий банк моделей для одновимірних періодичних структур є не повний. Незважаючи на простоту конструкції цих структур, не варто допускати хибне враження про простий набір їхніх властивостей. Їх широко використовують як основу для побудови елементної бази різноманітних пристроїв інфокомунікаційних систем, зокрема: фільтрів, антен, мультиплексорів та демультиплексорів тощо. Тому ця тематика має практичне значення та є актуальною.

Мета роботи – розроблення алгоритму на основі математичної моделі для створення програмного продукту, що дає змогу здійснювати розрахунок просторового розподілу електромагнітного поля одновимірної періодичної діелектричної структури.

Вихідні дані та результати. Одним із представників широкого класу одновимірних періодичних структур є діелектрична пластинка, діелектрична проникливість якої модульована однією періодичною послідовністю прямокутних функцій. Ця структура характеризується корисними інфокомунікаційними властивостями [2]. Для досягнення поставленої мети використаємо цю структуру та її математичну модель, яку запропоновано в роботі [3]:

$$F(\chi') = -2i \frac{\xi_0(\chi') / (\xi_0(\chi') - Z_0)}{1 - Z_M \sum_{n=-\infty}^{n=\infty} \frac{\text{sinc } P_n}{\xi(\chi' - nT) - Z_0}} \quad (1)$$

¹ ст. викл. І.В. Ничай, канд. техн. наук – НУ "Львівська політехніка"