ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.02+618.3

ІНТЕЛЕКТУАЛЬНІ МЕТОДИ ТА ЗАСОБИ ЕКСПЕРТНИХ СИСТЕМ МЕДИЧНОЇ ДІАГНОСТИКИ

ВАСИЛЕНКО О.О.

Досліджуються процеси моделювання діагностики клінічної медицини, розробка засобів аналізу даних і синтезу медичних знань та створення експертних систем клінічної медицини. Розглядаються методи застосування потенціалу штучного інтелекту в медицині та проведення діагностики на основі байєсівських мереж. Реалізовані інтелектуальні обчислювальні засоби у вигляді медичної експертної системи на підставі моделей та запропонованих у дослідженні засобів діагностування.

Ключові слова: медичні експертні системи, діагностика, медична база знань, синдроми, симптоми.

Key words: medical expert systems, diagnosis, medical knowledge base, syndromes, symptoms.

Вступ

В Україні офіційно зафіксовано перевищення кількості померлих над числом народжених. У 2017 році цей показник складав 47 осіб – тобто на кожні 100 померлих приходилося лише 53 новонароджених. Крім того, відповідно дослідженню, яке було проведено у США у 2012 році, внаслідок неправильної постановки діагнозу у реанімації щорічно гине 4050 пацієнтів. Отже зараз в Україні є дуже велика проблема зберігання життя людини. Проблема гострих запальних захворювань верхніх дихальних шляхів, гострого риносинуситу (ГРС) зокрема, є однією з найактуальніших у сучасній клінічній медицині. Останнім часом відзначають зростання частоти захворювань носа, зокрема навколоносових пазух. Це проявляється збільшенням як абсолютних (захворюваності та поширеності), так і відносних (частки у структурі оториноларингологічної патології). В Україні поширеність гострих ринітів, риносинуситів та ринофарингітів досягла 489,9 випадків на 10000 населення, а захворюваність – біля 5 – 15 випадків на 1000 населення залежно від сезону. Такі хворі становлять 60–65% амбулаторних пацієнтів оториноларингологів.

Для вирішення згаданих вище проблем необхідно провести відповідні дослідження підвищення якості лікування. Зокрема, треба розвивати напрямок дослідження діагностування ГРС якісного, без похибок та з високим рівнем клінічного мислення лікаря. Діагноз ГРС встановлюють на підставі клінічних даних. Висновок лікаря спирається на скарги пацієнта, анамнез, симптоми та ознаки захворювання, дані лікарського огляду та включає суб'єктивну оцінку тяжкості хвороби самим пацієнтом. Отже, дуже важливо, щоб на ранніх стадіях захворювання мати найбільш точний діагноз. Це можливо при наявності у лікаря інтелектуальної системи проведення первинного огляду хворого та виявлення анамнезу. Взагалі лікар приймає рішення стосовно діагнозу ГРС в умовах багатомірної невизначеності ознак хвороби, симптомів і багатокритеріальності вимог до процесу діагностування, що і підтверджує актуальність даного дослідження.

Крім того, за наказом Міністерства охорони здоров'я України від 11.02 2016 року за №85 було введено «Уніфікований клінічний протокол первинної, вторинної (спеціалізованої) та третинної (високоспеціалізованої) медичної допомоги: гострий риносинусит**».**

У Європі протягом довгого часу значення риносинуситів недооцінювалося. У дослідженні Європейського союзу для 57128 респондентів з 12 країн, відповідно до критеріїв EPOS, виявилося, що хронічним риносинуситом страждають 10,9% респондентів. У Голландії захворюваність риносинуситом склала 1880 на 100000 населення. Серед жінок вона була вище, ніж серед чоловіків, – 2310 проти 1440100000 населення. У більшості досліджених пацієнтів (69,0%) був зареєстрований один епізод риносинуситу, у 19,0% – один рецидив і лише у 12,0% – 2 і більше. В США захворюваність риносинуситом за останні 20 років зросла багаторазово. Про це свідчить, зокрема, динаміка кількості щорічних відвідувань лікаря: якщо в 1995 р. воно становило близько 12 млн, то до 2004 зросло до 32 млн.

Діагностику РНС на даний момент можна назвати одним з найважливіших напрямків у медицині. Раннє виявлення патологічних станів – запорука успішного лікування. Будь-яке захворювання легше лікувати, коли визначені симптоми та сформульований діагноз на ранніх стадіях. Отже, якщо з'являється можливість раніше виявити стан хвороби, то можна запобігти її розвитку.

При встановленні діагнозу за симптомами необхідно виділити суб'єктивні і об'єктивні ознаки патології. До суб'єктивних ознак захворювання, які слід виділяти за відчуттями пацієнта, це будь-які патологічні зміни в організмі. До об'єктивних ознак хвороби відносять будь-які відхилення від норми на основі лише фізикального та інструментального видів досліджень, які лікар може оцінювати самостійно, не вдаючись до спілкування з пацієнтом.

Класична тактика діагностування складається з декількох етапів:

– збір анамнезу (anamnesis vitae і anamnesis morbi);

– огляд;

– лабораторне обстеження (аналізи);
– інструментальне обстеження.

Об'єкт дослідження – процес отримання та обробки медичних даних, знань та засобів діагностування ГРС людини за допомогою медичних інтелектуальних систем.

Предмет дослідження – медична діагностична інтелектуальна система, алгоритми прийняття рішень, математичні моделі діагностування, засоби зберігання біомедичних Big Data та програмна реалізація системи.

Актуальними завданнями для дослідження є необхідність:

1. Провести системний аналіз моделей і методів пошуку діагнозу ГРС.
2. Вибрати методи представлення релевантних та інформативних даних та знань про пацієнтів і хвороби, тобто розробити онтологічні чи прецедентні бази даних та знань на підставі аналізу текстів з Інтернету, огляду пацієнта та клінічних записів щодо його захворювань.
3. Розробити математичні алгоритми діагностики типу (характеру) хвороби з використанням методів інтелектуального аналізу даних (Data Mining, моделі на семантичних мережах представлення зв'язків симптомів та ознак хвороб, логіко-статистичних методів аналізу і синтезу).
4. Розробити систему збору даних та знань для прийняття рішень щодо вибору клінічних діагнозів з використанням онтологій та прецедентів з предметної області.
5. Розробити архітектурні і структурні рішення для надійного збереження та керування у великих медичних базах даних та знань.
6. Розробити медичну експертну систему та засоби її використання при діагностуванні, в умовах невизначеності нечітких даних.

Постановка завдання дослідження

Постановка завдання дослідження медичної експертної системи (МЕС) може бути сформульована в загальному вигляді так:

$$\mathrm{MEC} = \langle \mathrm{T}, \mathrm{D}(\mathrm{Si}, \mathrm{O}), \mathrm{SP}(\mathrm{EXD}, \mathrm{ME}) \rangle, \qquad (1)$$

де $\mathrm{D}(\mathrm{Si}, \mathrm{O}) \Leftrightarrow \underset{\mathrm{T},\Re}{\mathrm{Opt}}(\mathrm{Si}, \mathrm{O})$, що має бути оптимізовано на множині симптомів і діагностичних ознак при мінімальних часових витратах пошуку діагнозу і рішень по діагностування хвороби на множені часу T – і мінімальної помилки діагнозу на множині ймовірностей помилок – $\Re$. При цьому повинні враховуватися як одинокі, так і комплекси симптомів і ознак, що вибираються з бази знань (БЗ) онтологій та прецедентів. Таким підходом на основі попереднього системного аналізу існуючиі прецедентні або онтологічні дані можуть бути зібрані в реляційну базу даних (БД), що будується за допомогою пошуку вербальної інформації в Інтернеті, її системного аналізу, застосування алгоритмів Data Mining, введення поняття «мікроситуації».

Модель МЕС в момент часу t :

$$\mathrm{MEC}' \to \mathrm{Mod}_{\mathrm{t}} \Leftrightarrow \underset{\mathrm{T},\Re}{\mathrm{Opt}}\left(\mathrm{Pr}'_1, \mathrm{Pr}'_2, \ldots, \mathrm{Pr}'_{\mathrm{j}}, \ldots, \mathrm{Pr}'_{11}\right), \qquad (2)$$

де $\mathrm{Pr}' = \{\mathrm{Pr}'_1, \mathrm{Pr}'_2, \ldots, \mathrm{Pr}'_{11}\}$ – сукупність статистичних процедур перетворення інформації про МЕС; Pr'_1 – процедура багатофакторного аналізу та ранжування параметрів МЕС; Pr'_2 – процедура регресійного аналізу статистичних прецедентних або онтологічних даних симптомів і ознак; Pr'_3 - процедура дисперсійного аналізу симптомів і ознак; Pr'_4 – процедура визначення діагнозу пацієнта з найвищою імовірністю достовірності; Pr'_5 – процедура Data Mining; Pr'_6 – складання дерева рішень; Pr'_7 – процедура з використанням нечіткої логіки нейронних мереж; Pr'_8 – процедура об'єднання результатів системного аналізу онтологій та прецедентів МЕС і аналізу стану пацієнтів EDR, пошуку в Інтернеті, виявлення даних, знань з текстової інформації про множини діагнозів $\{{}^{\mathrm{Si}}\mathrm{D}^{\mathrm{O}}\}$, симптомів $\{\mathrm{Si}\}$ і ознак $\{\mathrm{O}\}$ захворювань та логічних законів клінічного мислення; Pr'_9 – процедура оцінки ризику неправильного діагнозу хворим на множинах $\{\mathrm{D}\}, \{\mathrm{Si}\}, \{\mathrm{O}\}$; Pr'_{10} – процедура визначення клінічного діагнозу з використанням множини «мікроситуацій» [1,2] $\{{}^{\mathrm{Si}}\mathrm{Msi}^{\mathrm{O}}\}$.

В результаті проведених досліджень та попереднього аналізу апріорних даних предметної області МЕС для пошуку діагнозу, симптому, діагностичної ознаки маємо узагальнену модель МЕС з урахуванням логічного зв'язку множини симптомів $\{\mathrm{Si}\}$, множини ознак $\{\mathrm{O}\}$ та мікроситуацій $\{{}^{\mathrm{Si}}\mathrm{Msi}^{\mathrm{O}}\}$ варіантами діагнозу захворювання $\{{}^{\mathrm{Si}}\mathrm{D}^{\mathrm{O}}\}$:

$${}^{\mathrm{Si}}\mathrm{D}^{\mathrm{O}}_{\mathrm{k}} = \begin{cases} {}^{\mathrm{Si}}\mathrm{Msi}^{\mathrm{O}}_{1,1} = \mathrm{f}\left(\mathrm{O}^{\mathrm{Si}_1}_{1,1}, \mathrm{O}^{\mathrm{Si}_1}_{2,1}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{i,l}}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{N,l}}\right) \\ \ldots \\ {}^{\mathrm{Si}}\mathrm{Msi}^{\mathrm{O}}_{1,2} = \mathrm{f}\left(\mathrm{O}^{\mathrm{Si}_1}_{1,2}, \mathrm{O}^{\mathrm{Si}_1}_{2,2}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{i,2}}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{N,2}}\right) \\ \ldots \\ {}^{\mathrm{Si}}\mathrm{Msi}^{\mathrm{O}}_{\mathrm{j,d}} = \mathrm{f}\left(\mathrm{O}^{\mathrm{Si}_1}_{1,\mathrm{d}}, \mathrm{O}^{\mathrm{Si}_1}_{2,\mathrm{d}}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{i,d}}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{N,d}}\right) \\ \ldots \\ {}^{\mathrm{Si}}\mathrm{Msi}^{\mathrm{O}}_{\mathrm{M,N}} = \mathrm{f}\left(\mathrm{O}^{\mathrm{Si}_1}_{1,\mathrm{N}}, \mathrm{O}^{\mathrm{Si}_1}_{2,\mathrm{N}}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{i,N}}, \ldots, \mathrm{O}^{\mathrm{Si}_1}_{\mathrm{N,N}}\right) \end{cases}, \qquad (3)$$

де $\mathrm{k} = \overline{1, \mathrm{M}}, \mathrm{d} = \overline{1, \mathrm{N}}$.

На відміну від аналогічних пошукових алгоритмів [1], в (3) маємо ранжовані діагнози ${}^{\mathrm{Si}}\mathrm{D}^{\mathrm{O}}_{\mathrm{k}}$, з яких лікар чи експерт обґрунтовано буде вибирати за рекомендацією інтелектуального алгоритму експертної системи той, який буде найближчим

за теорією корисності [5] до онтологічного або прецедентного еталону.

Для виявлення деталей розробки БЗ маємо n хворих у контрольній групі, у яких встановлено m діагнозів. Серед n хворих мають діагноз $i=\overline{1,M}$, тобто:

Kl_1 – має діагноз 1;

Kl_2 – має діагноз 2;

…

Kl_g – має діагноз i;

…

Kl_G – має діагноз M;

причому деякі хворі можуть з певною ймовірністю мати як діагноз ${}^{Si}D_k^O$, так і діагноз ${}^{Si}D_j^O, k \neq j$, таким чином, має місце нерівність: $M > G$.

Розглянемо діагноз D_j^O. За результатами обстеження у $Kl_g, g=\overline{1,G}$ хворих є симптоми:

$$Kl_{D_j^O}(Si_1, Si_2, \ldots, Si_k, \ldots, Si_s),$$

де Si_k – набір диференційно-діагностичних симптомів хвороби D_k. Утворюється вектор симптомів:

$$Si_k = \{Si_1, Si_2, \ldots, Si_s\}.$$

Симптоми, в свою чергу, належать підгрупі деякої групи ознак (3). Наприклад, група ознак «загально клінічні дослідження» буде включати в себе такі підгрупи: клінічний аналіз крові з лейкоцитарною формулою і ШОЕ, копрограму, аналіз калу на цисти і вегетативні найпростіші s форми, загальний аналіз мокротиння та інші. Група «біохімічні аналізи» буде включати в себе підгрупи: біохімічний аналіз крові, білірубін, глюкозу в плазмі, білок загальний в сироватці та інші.

Розрахуємо частоту ознак, які визначаються емпіричним шляхом для вибірки пацієнтів, і підрахуємо, з якою частотою зустрічаються симптоми для кожного. Відзначимо, що чим більше буде обсяг вибірки, тим точніший буде показник частоти, з якою зустрічаються симптоми.

Потім для контрольної групи хворих з діагнозом D_j^O розрахуємо ваговий коефіцієнт $w_{D_o^O}$:

$$w_{D_j^O} = \frac{f_{D_j^O}({}^{Si}Msi^O)}{\sum f_j\left({}^{Si}Msi^O\right)} \cdot 100\%, \qquad (4)$$

де $f_{D_j^O}\left({}^{Si}Msi^O\right)$ – частота зустрічальності f_q, %, що визначена на множині виявлених з медичної практики або Інтернету онтологічних чи прецедентних даних або знань, яка функціонально залежить від так званих мікроситуацій, що ранжуються по впливу на вибір при прийнятті рішень щодо діагнозу, які виявляються процедурою Pr_{10} на множинах симптомів та ознак захворювань, що розраховуються, як вказано в (3).

Далі побудуємо еталонний вектор патології для хвороби D_j^O, логічної за вибраними критеріями. Осередок вектора має значення true, якщо дане захворювання має цей симптом, і значення false інакше [3].

Тут можливо використати простий варіант статистичного аналізу, який було запропоновано в [1], куди входять еталонний вектор хвороби D_j^O, частота виявлення й ваговий коефіцієнт кожного симптому без врахування статистичного аналізу впливу симптомів на діагноз [1,4].

Таким чином, будь-якому діагнозу D_j^O відповідає певний набір симптомів – ознак (3), які можна представити у вигляді вектора відповідності $B_{D_j^O}$, де «1» і «0» ставиться у відповідність «+» і «-»:

$$B_{D_j^O} = B_{D_j^O}\left(\delta_{D_1^O}, \delta_{D_2^O}, \ldots, \delta_{D_{k+t}^O}\right),$$

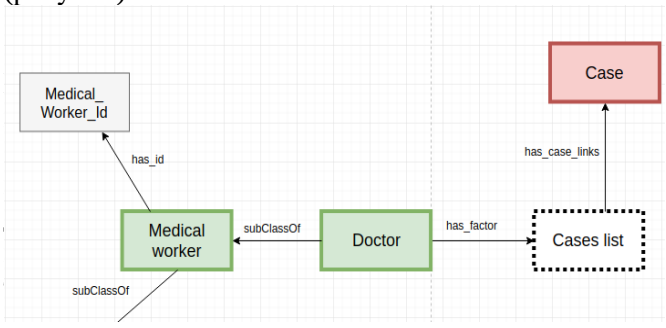
тут

$$\delta_{i,j} = \begin{cases} 1, – \text{якщо це захворювання має даний симптом } Si_k \\ 0, – \text{інакше.} \end{cases}$$

Цей простий, відомий засіб статистичного аналізу підтверджує правильність напрямку даного дослідження і може бути надалі використовуватися у подальшому розвитку засобів діагностування.

Моделювання

У досліджені були побудовані моделі онтології [6]. Насамперед, було розроблено модель пацієнта, яка має відповідні класи, атрибути і зв'язки (рисунок).



Підклас Medical Worker головного класу Patient

Було розглянуто 3 найбільш показові експертні системи, кожна з яких має свої унікальні особливості. Жодна з представлених на даний момент систем не може напряму аналізувати анамнез пацієнта, а тільки через запитання. Кожній системі не достає індивідуального підходу до пацієнта. Результати аналізів повинні також наводитися у вже обробленій лікарем формі, типу "Підвищена глюкоза у крові", а не "Кількість глюкози в аналізі крові пацієнта".

На даний момент не існує такого штучного інтелекту, який міг би в собі вмістити простоту використання “Перевірщика симптомів” для простого користувача, повноту “Інструмента диференційного діагностування” лікаря та систему обліку медичних карток пацієнтів. Вже існує багато “Перевірщиків симптомів”, результативність яких не дуже велика, а усі програми діагностування для професійних лікарів не можуть брати до уваги та вести медичні картки пацієнтів, що доступно системам обліку. Таким чином, з’являється ніша професійних інструментів діагностики, які також можуть аналізувати та допомагати вести лікарю медичну картку.

Розробка вимог щодо проекту

Вимоги для поведінки системи

Проаналізувавши предметну область, проблеми поставленої задачі та аналогічні рішення, можна скласти список випадків використання (табл. 1) (“Use cases”), які програмний додаток повинен виконувати.

Таблиця 1. Use cases

Дії користувача	Дії програми
а) Перехід у вкладинку “База знань” б) Ввід пошукового терміну у спеціальному полі в) Натискання на кнопку “Пошук”	а) Пошук вершин за назвою серед хвороб, синдромів, симптомів, яка сходиться з пошуковим запитом б) Повернення статей, які містять опис шуканого терміну
а)Перехід у вкладинку “Клінічний аналіз” б) Ввід симптомів в) Натискання кнопки “Діагностика”	а)Передача списку вибраних підтверджених симптомів на сервер б) Знаходження хвороб, які визивають дані симптоми в) Підлік вірогідностей хвороб г)Видача висновку роботи програми, який містить список хвороб та їх вірогідності, а також запитання на підтримку наступного опорного симптому
а)Перехід у вкладинку “Екстрений аналіз” б) Ввід симптомів та поведінки пацієнта в) Натискання кнопки “Діагностика”	а)Видавання допоміжного списку симптомів користувачеві б) Передача списку вибраних підтверджених симптомів на сервер в) Знаходження хвороб та алгоритмів дії для допомоги пацієнту г)Видача користувачу знайдених хвороб з їх вірогідностями та алгоритмами дії, а також запитання для підтвердження наступного симптому
а) Перехід у вкладинку “Клінічний аналіз” б) Ввід симптомів в) Натискання кнопки “Діагностика” г) Відповідь на запитання “так”	а) Додається підтверджений симптом у список фактів б) Передача списку підтверджених симптомів на сервер в) Знаходження хвороб, які визивають дані симптоми
	а) Підрахунок вірогідностей хвороб б) Видача висновку роботи програми, який містить список хвороб та їх вірогідності, а також запитання на підтримку настпного опорного симптому
а) Перехід у вкладинку “Екстрений аналіз” б) Ввод симптомів та поведінки пацієнта	а) Додається підтверджений симптом у список фактів
б) Натискання кнопки “Діагностика” в) Відповідь на запитання “так”	б) Передача списку вибраних підтверджених симптомів на сервер в) Знаходження хвороб та алгоритмів дії для допомоги пацієнту
	Видача користувачу списку знайдених хвороб з їх вірогідностями та алгоритмами дії, а також запитання для підтвердження наступного симптому
а) Перехід у вкладинку “Клінічний аналіз” б) Ввід симптомів в) Натискання кнопки “Діагностика” г) Відповідь на запитання “ні”	а) Додається заперечений синдром у список фактів б) Передача списку фактів на сервер в) Знаходження хвороб, які визивають дані симптоми г) Підрахунок вірогідностей хвороб д) Видача висновку роботи програми, який містить список хвороб та їх вірогідності, а також запитання на підтримку наступного опорного симптому
а) Перехід у вкладинку “Екстрений аналіз” б) Ввід симптомів та поведінки пацієнта в) Натискання кнопки “Діагностика” г) Відповідь на запитання “ні”	а) Додається запереченний симптом у список фактів б) Передача списку вибраних підтверджених симптомів на сервер в) Знаходження хвороб та алгоритмів дії для допомоги пацієнту г) Видача користувачу знайдених хвороб з їх вірогідностями та алгоритмами дії, а також запитання

Продовження табл. 1

Дії користувача	**Дії програми**
	для підтвердження наступного симптому
а) Перехід у вкладинку “Клінічний аналіз” б) Ввод симптомів	а) Додання ідентифікатора користувача у список фактів
Дії користувача	Дії програми
Додавання користувача Натискання кнопки “Діагностика”	б) Передача списку вибраних підтверджених симптомів на сервер в) Знаходження хвороб, які викликають дані симптоми г) Знаходження факторів, які вчитуються в аналізі з анамнезу користувача д) Підрахунок вірогідностей хвороб е) Видача висновку роботи програми, який містить список хвороб та їх вірогідності, а також запитання на підтримку наступного опорного симптому
а) Перехід у вкладинку “Екстрений аналіз” б) Ввід симптомів в) Додавання користувача г) Натискання кнопки “Діагностика”	а) Передача списку вибраних підтверджених симптомів на сервер та ідентифікатора користувача б)Знаходження хвороб та алгоритмів дії для допомоги пацієнту з врахуванням його анамнезу
	в) Видача висновку роботи програми, який містить список хвороб та їх вірогідності, а також алгоритми дії та запитання на підтримку наступного опорного симптому

Обґрунтування обраних технологій та конфігурація

Розроблений кієнт-серверний додаток складається із двох основних частин:

– клієнтської частини, що відповідає за отримання даних від користувача та коректний трансферінг цих даних на сервер;

– серверної частини, що відповідає за операції над даними, відповідно до бізнес-логіки програми, що включає в себе коректну роботу з базою даних та відправлення оброблених даних назад до клієнта.

Але для оптимальної роботи програми окрім ефективного взаємозв’язку цих двох компонентів дуже важливим є прийняття архітектурного рішення стосовно бази даних.

Зважаючи на бізнес-логіку даного проекту, можна зробити припущення щодо ефективності графових баз даних, наприклад Neo4j. Це пояснюється тим, що є постійна потреба у зв’язках між таблицями симптомів та припустимими діагнозами, що може бути неоптимальним у реляційних базах даних. У процесі формування діагнозу необхідно використовувати ітерацію елементів залежно від типу їх зв’язку. Наприклад, симптоми та діагнози можна представити у вигляді нодів у графі. У процесі дослідження було порівняно реляційні бази даних та базу даних Neo4j (табл. 2).

Таблиця 2. Порівняння реляційних баз даних та Neo4j

Критерій	**Реляційна база даних**	**Графова база даних Neo4j**
Зберігання даних	Зберігання даних у фіксованих, попередньо визначених таблицях з рядками та стовпцями з підключеними даними часто не перетинаються між таблицями, що знижує ефективність запитів	Структура зберігання графів з неспорідненою сусідністю приводить до більш швидких транзакцій і обробки відносин даних
Моделювання даних	Модель бази даних повинна бути розроблена з моделями та переведена з логічної моделі на фізичну. Оскільки типи даних і джерела повинні бути відомі заздалегідь, будь-які зміни вимагають тижня простою для впровадження	Гнучка модель даних без розбіжності між логічною та фізичною моделлю. Типи даних і джерела можуть бути додані або змінені в будь-який час, що приводить до різкого скорочення часу розробки і справжньої швидкої ітерації
Ефективність запиту	Продуктивність обробки даних страждає з кількістю та глибиною JOIN (або запитаних відносин)	Обробка графів забезпечує нульову затримку та продуктивність у реальному часі, незалежно від кількості або глибини відносин
Мова запиту	SQL: мова запитів, що збільшується по складності з кількістю JOIN, необхідних для підключених запитів даних	Cypher: рідна мова запиту на графік, який забезпечує найбільш ефективний і виразний спосіб опису запитів відносин

Продовження табл. 2

Критерій	Реляційна база даних	Графова база даних Neo4j
Підтримка транзакцій	Підтримка транзакцій ACID, необхідна для корпоративних додатків для послідовних і надійних даних	Зберігає транзакції ACID для цілком послідовних і надійних даних цілодобово – ідеально підходить для глобальних корпоративних додатків
Ефективність центру обробки даних	Консолідація серверів можлива, але дорога для розширення архітектури. Масштабувати архітектуру є дорогим з точки зору купівлі, використання енергії та управління часом	Дані та відносини зберігаються як і раніше разом із підвищенням продуктивності, коли складність і масштабність зростають. Це приводить до консолідації серверів і неймовірно ефективного використання апаратних засобів

Отже, обрана реляційна архітектура – це рішення для тих випадків, коли все просто і однозначно, але це неповоротка архітектура для створення складних і гнучких запитів, обробки різноманітних і багаторазових зв'язків між об'єктами. Однак не можна забувати про такі переваги SQL-баз даних, як можливість створення складних (JOIN) запитів. Такий підхід робить стандартизовані реляційні БД більш універсальними.

Основною перевагою графових баз даних є універсальність, адже в них можна зберігати і реляційні, і документарні і складні семантичні дані. А сама модель побудови БД може змінюватися і модифікуватися в процесі розвитку додатку без зміни архітектури і вихідних запитів.

З іншого боку, при незначній кількості зв'язків і великих обсягах даних графові БД демонструють значно нижчку продуктивність, і це потрібно обов'язково мати на увазі. Ще одним важливим обмеженням є те, що в даний момент практично не існує графових баз даних, які б добре працювали в паралельних архітектурах.

Таким чином, для розробки програми було вирішено інтегрувати додаток із Neo4j базою даних. Також вважається, що для роботи із додатком достатнім є консольний інтерфейс.

Для розробки серверної частини програми було обрано мову програмування Java. Для автоматизації зборки проекту було обрано фреймворк Apache Maven. Apache Maven - інструмент управління проектами програмного забезпечення. Базуючись на концепції об'єктної моделі проекту (POM), Maven може керувати складанням проекту, звітністю та документацією з центральної частини інформації.

По-перше, для використання переваг інверсії залежностей було вирішено використовувати фреймворк Spring. Spring – це технологія з відкритим кодом, що дозволяє значно спростити процес впровадження залежностей та тестування додатку на Java.

Конфігурації, що було використано для налаштування Spring та Spring Boot, у проекті можна розглянути, де у файл pom.xml додано потрібні «dependency».

Neo4j пропонує багатий набір можливостей інтеграції для розробників, які використовують Java або інші мови JVM. Автономний сервер Neo4j може бути встановлений на будь-якій машині, а потім доступ до нього за допомогою двійкового протоколу «bolt».

Для інтегрування Neo4j embedded-бази даних було використано Spring Data Neo4j бібліотеку від розробника Neo4j. Spring Data Neo4j тісно інтегрується з Spring Framework і пропонує об'єктно-графічне відображення поверх Neo4j.

Функції (на основі Neo4j-OGM):

– Spring інтеграція,

– мапінг графа об'єктів на основі анотування,

– підтримка сховища на основі інтерфейсу з анотаційними та отриманими методами пошуку,

– швидке сканування метаданих класів,

– оптимізоване управління завантаженням даних та відстеження змін для мінімальної передачі даних,

– кілька транспортів: двійковий протокол, HTTP.

Розробка Java-бізнес-додатків часто вимагає відображення багатьох моделей у базу даних. Бібліотека Neo4j-OGM є чистою бібліотекою Java, яка може зберігати (анотовані) об'єкти домену за допомогою Neo4j. Він використовує оператори Cypher для обробки цих операцій у Neo4j. Object-graph-mapping (OGM) підтримує зміни відстеження, щоб мінімізувати необхідні оновлення та транзитивне збереження (читання та оновлення сусідів об'єкта).

Підключення до Neo4j здійснюється за допомогою драйвера, який може використовувати двійковий протокол, вбудовані інтерфейси API HTTP або Neo4j. Інтеграція із Neo4j також потребує конфігурації у Java-класі із використанням анотацій із пакету org.neo4j із щойно імпортованої бібліотеки.

Окрім налаштування середовища для зберігання embedded-бази даних Neo4j, у конфігураційному

файлі є налаштування sessionFactory та transactionManager бінів фреймворка Spring, що потрібно для впровадження залежностей.

Таким чином, засобами імпортованих бібліотек було створено ноди графа. Приклад із програмного коду – домен пацієнта, який бере до уваги основні дані щодо хворого: вік, стать, історію подорожей та анамнез.

Таким чином, база даних стала готовою для добавлення усіх даних з метою детального аналізу синуситу.

Розробка бази знань

На основі сформульованих вимог до програмного додатку в аналізі предметної області сутностей та зв'язків між ними, а також діаграм потоків було з'ясовано примірну структуру бази даних.

Підграф користувача складається з вершин користувача, сторінки анамнезу та його причини. Серед цих вершин є також зв'язки "Має сторінку анамнезу" (has_anamnesis_page) та "Приклад причини" (instance_of_cause).

Вершина мережі користувача має такі ярлики:

а) Person – визначає, що вершина належить до користувача;

б) Patient – визначає, що користувач має доступ до функціоналу пацієнта;

в) Doctor – визначає, що користувач має доступ до функціоналу лікаря;

г) Administrator – визначає, що користувач має доступ до функціоналу адміністратора.

Вона також може мати виходяще ребро "has_anamnesis_page", яке входить у вершини сторінки анамнезу. Вершина сторінки анамнезу має в собі ярлик "AnamnesisPage".

Вершина прикладу причини з'єднана входящим в неї зв'язком "instance_of_cause" з вершини сторінки анамнезу та має такі ярлики:

а) Cause – визначає, що вершина належить до типу вершин-причин, які враховуються у алгоритмі підрахунку вірогідності;

б) AnamnesisCause – визначає, що вершина належить до типу вершин-причин з анамнезу пацієнта;

в) KnowledgeItem – визначає, що вершина має при собі опис і належить до знань про предметну область.

Хвороба – це порушення нормальної життєдіяльності організму, обумовлене функціональними або (і) морфологічними змінами і повинна розглядатися у повному комплексі ефектів, що вона породжує. За це відповідає семантичний граф, який складається з вершин-причин, які описують порушення в нормальній роботі організму, які є ефектами тих чи інших хвороб, тобто симптоми та синдроми. Представлені у табл. 3 інформаційні об'єкти діагностування захворювань перебувають у певних ієрархічних, логічних стосунках один з одним таким чином, що їх реквізити є джерелами один до одного.

Таблиця 3. Діагностика патологічних симптомів

Інформаційний об'єкт	**Категорія баз даних**	**Основні реквізити**	**Призначення**
Симптоми	Таблиця	Код симптому. Вербально-формалізований опис синдрому	Ввід даних
Симптоми, для яких виконуємо діагностику	Таблиця	Код синдрому. Назва синдрому	Діагностика
Симптомокомплекси, для яких виконуємо діагностику	Таблиця	Код симптому, код синдрому. Діагностичні коефіцієнти	Діагностика
Випадок, для якого виконуємо діагностику	Таблиця	Код симптому. Вербально-формалізований опис синдрому	Ввід даних
Діагностика	Запит на вибірку у вигляді загальної таблиці	Назва синдромів ранжування за ступенем важливості Сумарні діагностичні коефіцієнти	Діагностика синдромів
Діалоговий інтерфейс вводу даних	Форма	Вербально-формалізований опис симптому	Ввід даних
Діалоговий інтерфейс аналізу даних	Форма	Назва синдромів ранжування за ступенем важливості. Сумарні діагностичні коефіцієнти	Інтерактивний аналіз даних
Медичний висновок	Звіт	Назва синдрому. Вербально-формалізований опис симптому. Діагностичні коефецієнти	Документування результатів

Діагностування

Ієрархічні відносини відображають етапи діагностичної процедури: введення даних, аналіз даних, діагностику синдромів та оформлення результатів.

Усі хвороби, що заперечують симптому, який аналізуємо, «вибувають» з подальшого аналізу. Якщо хвороба ніяк не заперечує симптому, то для неї обчислюється вірогідність з урахуванням підтверджених лікарем фактів [2,4,6]. Потім діагноз хвороби, яка має найбільшу вірогідність за симптомами і містить підтверджуюче запитання, буде відправлено лікарю для підтримки діалогу. Визначені хвороби та наступне запитання з ідентифікатором хвороби буде відправлено лікарю для підтримання діалогу.

В даному проекті доцільно застосовувати систему управління бізнес-правилами Drools. Ця система управління бізнес-правилами є інформаційною, яка використовується для супроводу, підтримки і виконання бізнес-правил аналізу. Основні переваги використання системи бізнес-правил:

1. Дозволяє описати складні алгоритми розгалуження, які, якщо були б написані за допомогою стандартного коду, були б складні в редагуванні.
2. Дозволяє розбити складне рішення на множенні правила.
3. Бізнес-логіка може оновлюватись досить швидко і не зупиняти додаток.
4. Модель може бути краще пояснена експерту, який не працює в технічній галузі.
5. Бізнес-логіка і дані розділені.
6. Краща масштабованість.

Основною функцією проекту є надання першої допомоги пацієнтам з урахуванням алергії і ускладнень. Для встановлення діагнозу необхідно за допомогою спеціального меню додати симптоми пацієнта, які можна отримати шляхом зовнішнього обстеження і опитування. Також для врахування алергій пацієнта необхідно ввести його id . Користувач отримає розгорнуту інформацію про пацієнта і його анамнез буде враховуватися при з'ясуванні діагнозу. Діагноз можна уточнювати, відповідаючи на додаткове пряме запитання, яке підтверджує або спростовує діагноз, який має високу вірогідність. Відповіді впливають на поставлений діагноз і з кожним разом стають все більш точними.

Введення вхідних даних здійснюється на підставі спостережень за пацієнтом і його клінічної картини. Симптоми являють собою набір пов'язаних параметрів з бази даних. Ієрархія діагнозів побудована за структурою міжнародної класифікації хвороб МКБ-10 і доповнена захворюваннями з медичних довідників. Додавання діагнозів відбувається через інтерфейс додавання діагнозу.

Після введення всіх даних необхідно натиснути кнопку «обробити», яка виконає аналіз прецедентів і дасть процентні співвідношення хвороб.

Як тільки всі симптоми і діагнози додані в списки з відповідними коефіцієнтами впевненості, лікар натискає на кнопку «Проконсультуватися» і МЕС автоматично переходить на сторінку перегляду результатів консультації.

Результати консультації відображаються у вигляді оновлених списків діагнозів і симптомів, а також самої інформації щодо міркувань у вигляді списку правил і запропонованих МЕС змін.

Висновки

Наукова новизна результатів роботи:

– вперше розроблено метод розподілу клінічної медичної ситуації на небезпечні і безпечні класи, який визначає найважливіші фактори впливу на клінічний стан пацієнта, що надає можливість за допомогою міри близькості мікроситуацій (з найбільшим впливом параметрів) запропонувати методику прогнозу стану пацієнта з використанням мереж довіри;

– вперше розроблено надійний метод резервування для зберігання біомедичних Big Data, який на практиці забезпечив високу надійність в порівнянні з існуючими методами;

– набув подальшого розвитку метод з використанням алгоритму Рене та побудови структури експертних систем для клінічної медицини, який відрізняється повторним використанням онтології вдалих результатів і дає можливість підвищити надійність і швидкість обробки вхідних даних, а також ефективність прийняття рішень;

– вдосконалена ситуаційна модель клінічної медицини для аналізу стану пацієнта, яка, на відміну від існуючих підходів, використовує ситуаційне уявлення кризової ситуації на основі трійки «лікар – онтології чи прецедент - пацієнт», що дозволяє прогнозувати небезпечні і безпечні ситуації для хворого швидше і з більшою точністю, ніж існуючі моделі.

Література:

1. *Kuzomin O., M. Ayaz Ahmad, Lyashenko V., N. Ameer Ahamad.* Formalization of avalanche climate to avalanche riskiness and avalanche safety classes in the emergency situations separation // International Journal of Advanced Research. 2015. Vol. 3, Issue 4. P. 684-691.

2. *Vasilenko O., Tolmachova T.* Development Of Intelectual Models And Methods Of Expert Systems Of Clinical Medicine / International Journal “Informattion Technologes&Knolegedge”. 2017. Vol. 11, N. 2. P. 186-199.

3. *Vasylenko O.* Data loss minimization in situation's centrums databases / Oleksandr Ya. Kuzomin, Oleksii Vasylenko. Global Risk Forum GRF Davos - Davos – Switzerland. 2014. P 153-154.

4. *Kuzomin O.Ya., Astapev O.O, Tolmacheva T.V.* Applying The Hits Algorithm On Web Archives // Proc. Of the Int. Conf. ITA in International Journal

"Informattion Models and Analyses". Bulgaria. Varna. Volume 6, Number 2. 2017. P. 119-131 ISSN 1314-6416 (printed), ISSN 1314-6432 (online).
5. *Kuzomin O., Vasylenko O.* Methods And Models For Building A Distributed Mobile Emergency Monitoring System // Proc. of the 17th International Multidisciplinary Scientific Geoconference SGEM 2017. Vol. 17. ISSUE 21 P. 433 – 440. ISBN 978-619-7408-01-0, ISSN 1314-2704. DOI: 10.5593/sgem2017/2.1.
6. *Kuzomin O., Stukin M., Bozhkov D.* Intelligent Geoinformatic Expert System For Providing Emergency Help During Extreme Situations // Proc. of the 18 International Multidisciplinary Scientific Geoconference SGEM 2018. Informatics Geoinformatics. Vol 18. Informatics, Geoinformatics and Remote Sensing. P. 269 – 292. ISBN 978-619-7408-40-9, ISSN 1314-2704. DOI: 10.5593/sgem2018/2.2.

Надійшла до редколегії 12.12.2018
Рецензент: д-р техн. наук, проф. Кузьомін О.Я.

Василенко Олексій Олександрович, керівник напрямку архітектури департаменту Аналітики та Big Data корпорації Toyota (Сідней, Австралія). Закінчив аспірантуру ХНУРЕ за напрямком «Інформаційні технології» у 2015 році. Проходив стажування в компанії GoGet Carshare Australia в м.Сідней, Австралія, у 2012-2013 р. за напрямком архітектура даних. Адреса: Україна, 61166, Харків, пр. Науки, 14.
Vasylenko Oleksii Oleksandrovich, Manager of Architecture for the Big Data and Analytics department, Toyota corp., Sydney, Australia. PhD of the 'Information Technology' stream NURE. Finished an internship at the GoGet Carshare Australia (Sydney, Australia) at 2012-2013 at the Data Architecture stream. Address: Ukraine. 61166. Kharkiv, Nauki Ave, 14.