

До основних тенденцій розвитку інформаційних систем та технологій можна віднести:

1. ускладнення інформаційних продуктів (послуг) у вигляді програмних засобів, баз даних і служб експертного забезпечення;
2. здатність до взаємодії;
3. ліквідацію проміжних ланок, яка дає можливість розміщувати замовлення безпосередньо за допомогою інформаційних технологій без посередників;
4. глобалізацію, яка націлена на отримання переваг за рахунок розподілу постійних і напівпостійних витрат на більш широкій географічній регіон;
5. конвергенцію, яка забезпечує відмінності між виробами та послугами, інформаційним продуктом та засобами, використанням в побуті і для ділових цілей, інформацією і розвагою, а також серед різних режимів роботи, таких як передача звукових, цифрових і відеосигналів.

Крім того, стисло позначимо до чого призводять тенденції розвитку інформаційних систем стосовно бізнесу:

1) здійснення розподілених персональних обчислень, коли на кожному робочому місці достатньо ресурсів для обробки інформації в місцях її виникнення;

2) створення розвинених систем комунікацій, коли робочі місця з'єднані для пересилання повідомлень;

3) розвитку гнучких глобальних комунікацій, коли підприємство включається у світовий інформаційний потік;

4) створення та розвитку систем електронної торгівлі;

5) усунення проміжних ланок у системі інтеграції «організація – зовнішнє середовище».

Таким чином, сучасний розвиток інформаційних систем та технологій дає змогу перейти на більш інтелектуальний рівень взаємодії користувача з програмним і апаратним забезпеченням.

ЕКСПЕРТНА СИСТЕМА «ВИБІР ВІДЕОКАРТИ»

Мартіросян М.К., Князєв І.А.

Науковий керівник – Петрова О.О., канд. тех. наук, доцент

Експертною системою (ЕС) називають систему підтримки прийняття рішень, яка містить знання з певної вузької предметної області, а також може пропонувати користувачеві рішення проблем з цієї галузі і обґрунтовувати їх. Експертна система складається з бази знань, механізму логічного виводу і підсистеми обґрунтувань.

Експертні системи призначені для відтворення досвіду, знань професіоналів високого рівня і використання цих знань в процесі управління. Вони розробляються з використанням апарату математичної логіки для експлуатації у вузьких областях застосування, оскільки їх використання вимагає великих комп'ютерних ресурсів для обробки і зберігання знань. В основі побудови експертних систем лежить база знань, яка ґрунтується на моделях представлення знань.

Перспективною для розробки експертних систем виявилася реалізація мови логічного програмування Visual Prolog, який використовує спрощений варіант синтаксису предикатної логіки, що забезпечує більш зрозумілий синтаксис, який є наближеним до природної мови. Однією з найважливіших особливостей Prolog є те, що, на додаток до логічного пошуку відповідей на поставлені питання, він може мати справу з альтернативами і знаходити всі можливі рішення. Замість звичайної роботи від початку програми до її кінця, Prolog може повертатися назад і переглядати більше одного «шляху» під час вирішення всіх складових завдання.

Мова логічного програмування Prolog має високий рівень абстракції, легкість та простоту в поданні складних структур даних, можливість моделювати логічні відношення та процеси, що істотно полегшує створення програми.

Розроблена мовою Visual Prolog експертна система призначена для визначення в залежності від потреб користувача необхідної комп'ютерної відеокарти за її характеристиками і має типові розділи Prolog–програми: Database, Predicates, Clauses. Під час розроблення експертної системи були розглянуті лише найпопулярніші типи відеокарт. У розділі Predicates було сформульовано 14 правил з відповідною рекомендацією по вибору відеокарт. Приклад одного з правил наведено на рисунку 1.

```
graphics_card_is(" Gigabyte GeForce RTX 2080 Ti XTREME 11G"):-
positive(graphics_card, " Recommended power supply from 750 W"),
positive(graphics_card, " Memory 11 GB"),
positive(graphics_card, " Memory type GDDR6"),
positive(graphics_card, " Tire width 352 bit"),
positive(graphics_card, " Video card length 290 mm"),
positive(graphics_card, " GPU frequency 1770 HMz"),
positive(graphics_card, " Memory frequency 14140 HMz"),
positive(graphics_card, " Occupied slots 2.7"),
positive(graphics_card, " Stream processors 4352"),
positive(graphics_card, " Texture blocks 272"),!.
```

Рисунок 1 – Приклад правила ЕС

Рекомендація експертної системи, яка отримана з застосуванням одного з розроблених правил, виведена на рисунку 2.

An expert system has been developed by students Knyazev Ivan and Martirosyan Mger

An Graphics Card Expert

Please answer the auestion 'yes' or 'no'.

graphics_card Recommended power supply from 750 W?yes
graphics_card Memory 11 GB?yes
graphics_card Memory type GDDR6?yes
graphics_card Tire width 352 bit?yes
graphics_card Video card length 290 mm?yes
graphics_card GPU frequency 1770 HMz?yes
graphics_card Memory frequency 14140 HMz?yes
graphics_card Occupied slots 2.7?yes
graphics_card Stream processors 4352?yes
graphics_card Texture blocks 272?yes

Graphics Card Gigabyte GeForce RTX 2080 Ti XTREME 11G.

Рисунок 2 – Рекомендації ЕС із вибору відеокарти для комп'ютера на підставі аналізу вимог для поставленого завдання

Розроблена експертна система націлена на допомогу у визначенні типу відеокарти трьом категоріям споживачів комп'ютерної техніки: офісним працівникам, геймерам та спеціалістам в галузі 3D-графіки, кожен з яких має свої вимоги до типу відеокарт. Рекомендації даної ЕС дозволять будь-якому користувачеві підібрати для себе найкращу відеокарту, яка буде підходити для його потреб.

АЛГОРИТМИ ПОШУКУ МАРШРУТІВ В ГРАФІ

Буров О.І.

Науковий керівник – Штельма О.М.

Базовою операцією в будь-якому графові алгоритмі є повний і систематичний обхід графа. Мета обходу – відвідати кожную вершину і кожне ребро рівно один раз в строго визначеному порядку. Існує два основних алгоритму обходу:

- пошук в ширину (breadth-first search – BFS);
- пошук в глибину (depth-first search – DFS).

Обидві процедури обходу графа використовують одну фундаментальну ідею – ми повинні позначити вершини, які вже бачили, щоб не намагатися відвідати їх знову. Інакше ми можемо зациклитися і ніколи не вийти з алгоритму. BFS і DFS розрізняють тільки порядком, в якому вони розглядають вершини.

Пошук в ширину слід використовувати в тому випадку, якщо:

- нам не важливий порядок, в якому ми обходимо вершини і ребра графа, тобто нас влаштує будь-який
- нам потрібно знайти найкоротший шлях в невваженому графі.