А. А. ЯРОВИЙ, Л. В. КРИЛИК, А. В. КОЗЛОВСЬКИЙ

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У СФЕРІ ШТУЧНОГО ІНТЕЛЕКТУ

Міністерство освіти і науки України
Вінницький національний технічний університет

А. А. ЯРОВИЙ, Л. В. КРИЛИК, А. В. КОЗЛОВСЬКИЙ

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У СФЕРІ ШТУЧНОГО ІНТЕЛЕКТУ

Електронний навчальний посібник
комбінованого (локального та мережного) використання

Вінниця
ВНТУ
2023

УДК[004.9+004.8](075.8)
Я76

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 3 від 28.09.2023 р.)

Рецензенти:
Н. І. Заболотна, доктор технічних наук, професор
Т. Б. Мартинюк, доктор технічних наук, професор
А. Я. Кулик, доктор технічних наук, професор

Яровий, А. А.

Я76 Сучасні інформаційні технології у сфері штучного інтелекту : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Яровий А. А., Крилик Л. В. , Козловський А. В. – Вінниця : ВНТУ, 2023. – 145 с.

В навчальному посібнику розглянуто основні теоретичні підходи сучасних інформаційних технології у сфері штучного інтелекту.

Навчальний посібник призначений для студентів спеціальності комп'ютерні науки та інших спеціальностей, в навчальному плані яких є аналогічна дисципліна.

УДК[004.9+004.8](075.8)

ЗМІСТ

ВСТУП

Світові тенденції розвитку інформаційних технологій характеризуються зростанням технічної досконалості та інтелектуальної насиченості. Інформаційні технології – це надзвичайно складна, багатогранна та багатопланова сфера людської діяльності.

За останні десятиліття передові країни світу інвестували сотні мільярдів доларів у розвиток технологій штучного інтелекту. Крім того, його використання в продуктах і послугах, а також в оборонному секторі стає одним із ключових елементів успіху на міжнародних ринках. В Україні штучний інтелект (ШІ) також розвивається у різних галузях промисловості. Особливу увагу приділено використанню штучного інтелекту у сфері кібербезпеки та оборони. Завдяки розвитку штучного інтелекту та його інтеграції з економічно важливими галузями можна суттєво збільшити частку інтелектуально насичених продуктів в Україні, щоб увійти в топ-10 світового рейтингу розвитку ШІ.

Інтелект − це дуже складна галузь знань, яку неможливо описати за допомогою тільки однієї теорії. Вчені будують цілу ієрархію теорій, що характеризують його на різних рівнях абстракцій.

На самому нижньому рівні цієї ієрархії знаходяться нейронні мережі, генетичні алгоритми та інші форми обчислень, які еволюціонують і дозволяють зрозуміти процеси адаптації, сприйняття, втілення та взаємодії з фізичним світом, що лежить в основі будь-якої форми інтелектуальної діяльності. На більш високому рівні абстракції розробники експертних систем, інтелектуальних агентів, систем розуміння природної мови намагаються визначити роль соціальних процесів у створенні, передачі і зміцненні знань.

Структура навчального посібника містить шість розділів. На думку авторів, матеріал розділів навчального посібника сприятиме якісному освоєнню теоретичних основ для ефективного застосування інтелектуальних інформаційних технологій у сфері штучного інтелекту в професійній діяльності для вирішення різноманітних завдань.

1 СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

1.1 Сучасні інтелектуальні інформаційні технології – зміна парадигми в контексті теоретико-методологічних основ наукового пошуку та педагогічної діяльності

У науково-технічній галузі термін «парадигма» своїм сучасним значенням зобов'язаний Томасу Куну та його книзі «Структура наукових революцій». Усталені системи наукових поглядів, в рамках яких проводяться дослідження, Кун назвав *парадигмами*, тобто в процесі розвитку наукової дисципліни одна парадигма може змінюватися іншою, водночас стара парадигма може існувати або навіть розвиватися певний час через те, що багато хто з її прихильників з тих чи інших причин виявляються **нездатними** адаптуватися до діяльності в іншій парадигмі.

«Під парадигмами я маю на увазі визнані всіма наукові досягнення, які протягом певного часу дають для наукового співтовариства модель постановки проблем та їх вирішення» (Т. Кун) [1].

Парадигма (дисциплінарна матриця) – це сукупність знань, методів, принципів дослідження та принципів цінностей, які є загальними для всіх членів наукової спільноти [1, 2].

Парадигма (у перекладі з грецької «приклад», «зразок») – у загальному розумінні – теоретико-методологічна модель [3].

За межами парадигми залишаються факти і теоретичні узагальнення, які не вписуються в існуючу парадигму. Накопичення знань (кумулятивний період) відбувається в період парадигмальної науки.

Розглянемо інші відомі означення.

• *Парадигма речення* – це система форм структурної схеми речення, в якій первинною формою є ядерне речення, а похідними від нього є його трансформації [3].

• *Парадигма* – це сукупність філософських, загальнотеоретичних засад науки; система понять і уявлень, характерних для певного періоду розвитку науки, культури і цивілізації [4].

• *Парадигма у мовознавстві* − вся сукупність форм слів, що утворюють лексему, а також зразок, схема словозміни [4].

• *Парадигма програмування* − система ідей і понять, що визначають стиль створення комп'ютерних програм, а також спосіб мислення програміста [5].

Термін «**Парадигма програмування**» вперше використав Роберт Флойд у своїй лекції лауреата премії Тюрінга, зазначивши, що в програмуванні ми можемо спостерігати явище, подібне до парадигм Куна, але, на відміну від них, парадигми програмування не є взаємовиключними:

«Якщо розвиток мистецтва програмування в цілому вимагає постійного винаходу та вдосконалення парадигм, то вдосконалення мистецтва окремого програміста вимагає від нього розширення свого репертуару парадигм».

За словами Роберта Флойда, на відміну від парадигм у науковому світі, описаних Куном, парадигми програмування можна комбінувати, збагачуючи інструментарій програміста [5].

Інформаційно-технологічна парадигма

Англійський економіст Крістофер Фрімен писав: «техніко-економічна парадигма – це концентрація взаємопов'язаних технічних, організаційних і управлінських інновацій, переваги яких потрібно шукати не тільки в новому асортименті продуктів і систем, але насамперед у динаміці відносної структури витрат на всі можливі інвестиції у виробництво. У будь-якій новій парадигмі будь-яку конкретну інвестицію або комбінацію інвестицій можна назвати «ключовим фактором» цієї парадигми, який характеризується спадом відносних витрат і широкою доступністю. **Сучасну зміну парадигми** можна розглядати як перехід від технології, що базується переважно на недорогих інвестиціях в енергетику, до технології, що базується переважно на недорогих інвестиціях в інформацію, отриманих завдяки прогресу в мікроелектроніці та телекомунікаційних технологіях» [6].

Розглянемо характеристики, що становлять суть парадигми інформаційних технологій. Разом вони утворюють основу інформаційного суспільства [6].

Перша характеристика – особливість нової парадигми полягає в тому, що інформація є її сировиною: у нас є технології, які впливають на інформацію, а не лише інформація, яка впливає на технології, як це було під час попередніх технологічних революцій.

Друга характеристика полягає у всеосяжності ефектів нових технологій. Оскільки інформація – це інтегральна частина деякої людської діяльності, всі процеси нашого індивідуального і колективного існування безпосередньо формуються (хоча, зрозуміло, не детермінуються) новим технологічним способом.

Третя характеристика полягає в мережевий логіці будь-якої системи або сукупності відношень, що використовує ці нові інформаційні технології. Схоже, що морфологія мережі добре пристосована до зростаючої складності взаємодій і до непередбачуваних моделей розвитку, які виникають з творчої міці таких взаємодій. Ця топологічна конфігурація – мережа – може бути тепер завдяки новим інформаційним технологіям матеріально забезпечена у всіх видах процесів і організацій. Без них мережева логіка була б занадто громіздкою для матеріального втілення. Однак ця мережева логіка потрібна для структурування неструктурованого за збереження вод-

ночас гнучкості, бо неструктуроване є рушійною силою новаторства в людській діяльності.

Четверта характеристика, пов'язана з мережевим принципом, але явно не належить тільки йому, полягає в тому, що інформаційно-технологічна парадигма основана на гнучкості. Процеси не тільки оборотні; організації та інститути можна модифікувати і навіть фундаментально змінювати шляхом перегрупування їх компонентів. *Конфігурацію нової технологічної парадигми відрізняє її здатність до реконфігурації* − вирішальна риса в суспільстві, для якого характерні постійні зміни і організаційна плинність. Поставити правила з ніг на голову, не руйнуючи організацію, стало можливим, оскільки матеріальну базу організації тепер можна перепрограмувати і переозброїти. Істотно, таким чином, зберігати дистанцію між оцінкою виникнення нових соціальних форм і процесів, індукованих та допустимих новими технологіями, і екстраполяцією потенційних наслідків таких подій для суспільства та людей: тільки конкретний аналіз й емпіричні спостереження зможуть визначити результат взаємодії між новими технологіями і виникаючими соціальними формами. Суттєво також ідентифікувати логіку, вбудовану в нову технологічну парадигму.

П'ята характеристика цієї технологічної революції − зростаюча конвергенція конкретних технологій у високоінтегровану систему, в якій старі, ізольовані технологічні траєкторії стають буквально нерозрізненними. Так, мікроелектроніка, телекомунікації, оптична електроніка і комп'ютери інтегровані тепер в інформаційних системах. У бізнесі, наприклад, існує (і ще деякий час буде існувати) відмінність чинів між виробниками і програмістами. Але навіть така диференціація розмивається зростаючою інтеграцією фірм у стратегічних союзах і спільних проектах, так само як і вбудовування програмного забезпечення в мікропроцесори. Більш того, в термінах технологічної системи один елемент неможливо уявити без іншого: мікрокомп'ютери визначаються переважно потужністю чипів, а проектування та паралельна обробка мікропроцесорів залежать від архітектури комп'ютерів. Телекомунікації є нині тільки однією з форм обробки інформації; технології передачі і зв'язку одночасно все ширше диверсифікуються та інтегруються в одній і тій самій мережі, де оперують комп'ютери.

Зміна парадигм

Термін «зміна парадигм» вперше введений істориком науки Томасом Куном для опису зміни базових посилань в рамках провідної теорії науки, тобто парадигми [1, 2].

Конфлікт парадигм, що виникає в періоди наукових революцій, − це, насамперед, конфлікт різних систем цінностей, різних способів вирішення

задач-головоломок, різних способів вимірювання і спостереження явищ, різних практик, а не тільки різних картин світу [1].

На думку Куна, для будь-яких парадигм можна знайти аномалії, які відкидаються у вигляді допустимої помилки або ж просто ігноруються і замовчуються. Кун вважає, що аномалії скоріше мають різний рівень значимості для вчених в окремо взятий час. Коли накопичується достатньо даних про значущі аномалії, що суперечать поточній парадигмі, згідно з теорією наукових революцій, наукова дисципліна переживає кризу. Протягом цієї кризи випробовуються нові ідеї, які, можливо, до цього не бралися до уваги або навіть були відкинуті. Зрештою, формується нова парадигма, яка набуває власних прихильників, і починається інтелектуальна «битва» між прихильниками нової парадигми і прихильниками старої.

Приклади змін парадигм в науці

- Зміна птолемеївської космології коперніковською.
- Об'єднання класичної фізики Ньютоном в зв'язний механістичний світогляд.
- Заміна максвеллівського електромагнітного світогляду ейнштейнівським релятивістським світоглядом.
- Розвиток квантової фізики, який перевизначив класичну механіку.
- Розвиток теорії Дарвіна про еволюцію шляхом природного відбору, що відкинув креаціонізм з позицій головного наукового пояснення різноманітності життя на Землі.
- Прийняття теорії тектонічних плит як пояснення великомасштабних геологічних змін.
- Прийняття теорії хімічних реакцій і окислювання Лавуазьє замість теорії флогістона (хімічна революція).
- Когнітивний напрямок у психології, що означав відхід від бихевіористського підходу до психологічних досліджень і перехід до вивчення когнітивних здібностей людини як головного чинника для вивчення поведінки, і трансперсональний рух, запропоновано новий погляд на надособистісний досвід і людський розвиток.
- Теорія Джеймса Лавлока про біосферу як єдину живу органічну систему.
- Заміна в теорії Дарвіна концепції синхронної еволюції на асинхронну.

Приклади змін парадигм в інформаційних технологіях

- Комунікаційні технології.
- Застосування електронних пристроїв високої швидкості та обчислювальної потужності.
- Сучасні високотехнологічні платформи.
- Технології зв'язку.
- Технології руху тощо.

Світові тенденції розвитку інформаційних технологій характеризуються зростанням їх технічної досконалості та інтелектуальної наповнюваності за рахунок створення в телекомунікаційних мережах високоінтелектуальних серверів із широким спектром інформаційних послуг, використання цифрової передачі аудіо- і відеоінформації, зростанням рівня використання оптичних систем та пакетних принципів передачі даних, широкосмугових радіосистем і супутникових каналів зв'язку.

Світ стрімко змінюється під впливом цифрових технологій: особистість окремої людини, соціум загалом, політика та економіка майже повністю залежить від них. Особистість і держава, інформаційні війни та тероризм, бізнес й економіка, технології та комунікації – оцифровано, здається, вже все. Виграє той, хто краще пристосується до цього «дивного нового світу» [7].

Розвиток технологій, особливо комунікаційних, поступово змінює наш світ. За останні роки кількість абонентів мобільного зв'язку та користувачів Інтернету збільшилась у багато разів, і цей процес продовжує набирати обертів. Змінюються моделі споживання інформації – слідом за ними маємо змінюватись й ми самі, а також наше уявлення про компанії, державні інститути, суспільство. Зокрема, окремі фахівці виділяють п'ять основних фундаментальних процесів, які характеризують суттєвий вплив революційного розвитку IT технологій у суспільстві, що також яскраво характеризує зміну парадигми, епоху якої дослідники визначають як «Новий цифровий світ» (рис. 1.1) [7, 8].

Рисунок 1.1 – Характерні ознаки при зміні парадигми в контексті «Нового цифрового світу»

Перша та найочевидніша зміна – стрімке зростання кількості можливостей завдяки новим цифровим технологіям. Багато ринків і корпорацій,

особливо в країнах, що розвиваються, відчувають цей ефект на собі. Простіше кажучи, можна досягати набагато більшого ціною менших зусиль.

Доступ величезної кількості користувачів до Мережі означає, що компаніям стає все простіше збирати та використовувати дані. Причому настільки простіше, що це змінює поняття конкуренції та ефективності, а перед клієнтами відкриває небувалі можливості. Наприклад, компанія Amazon, аналізуючи статистику продажу та іншу інформацію, може пропонувати своїм покупцям позики. Це стає гарною підмогою для тих клієнтів, які не можуть собі дозволити взяти кредит у звичайному банку.

Останнім часом точиться дуже багато дискусій навколо такої відносно нової технології як 3D-друк. 3D-друк став гарним способом підвищити продуктивність праці для багатьох компаній. Деякі дослідники припускають, що в майбутньому буде популярна така модель: клієнт передає через Інтернет специфікації, відповідно до яких оператор тривимірного принтера виготовляє продукцію в замовленому обсязі. «Це не замінить гектари виробничих площ, які зараз використовуються під серійне виробництво в багатьох галузях, але безпрецедентно розширить асортимент продуктів, доступних жителям розвинених країн», – пишуть Шмідт і Коен [7].

Поява нововведень у сфері комунікаційних та інших технологій призводить до того, що життя організацій стає більш цікавим та насиченим. «Вражає те, як можуть змінити світ навіть незначні успіхи в розвитку технологій, якщо при цьому зростають можливість доступу до мережі та взаємозалежність мешканців різних країн», – пишуть автори. Зокрема, спілкування з клієнтами та співробітниками з інших міст та країн настільки спростилося, що сьогодні нікого не здивуєш аутсорсингом робочої сили чи отриманням замовлення з іншого кінця світу. Далі цей процес лише поглиблюватиметься. У майбутньому для організацій стане звичним явищем тримати команду продавців в одній країні, HR-відділ – в іншій, а R&D – у третій.

Платформи для віддаленої взаємодії стають все більш досконалими, тому незабаром найсерйознішою перешкодою для ефективної комунікації стануть культурні відмінності, а не мовний бар'єр чи відсутність технологій [7, 8].

Як підсумок, потрібно зазначити, що розвиток ІТ-технологій та доступу до Інтернету породжують цілу низку як складних процесів, що потребує підвищеної уваги до кібербезпеки, захисту особистих даних, боротьби з фейками, наклепами, інформаційним тероризмом, тощо (що є негативною складовою), так і допомагають боротися зі зловживаннями, стражданням та насильством, забезпечують прозорість рішень влади, покращують здоров'я і досуг людини, підвищують ефективність праці та забезпечують нові можливості, що є безперечно вагомою позитивною складовою. Як зазначають окремі дослідники, – найкраще, що всі ми здатні зробити для підвищення якості життя на планеті, це забезпечити людям доступ до мережі та

дати їм можливість користуватися благами технологій. «Надайте їм доступ, а решту вони зроблять самі». Однак не потрібно забувати про те, що нові технології – це завжди нові виміри складності світу, в якому ми живемо. І для того, щоб виграти від їхньої появи, компаніям, людям та державам доведеться чимало попрацювати [7, 8].

1.2 Науково-методичні основи і стандарти в галузі інформаційних технологій та штучного інтелекту

Інформаційні технології – це надзвичайно складна, багатоаспектна та різнопланова сфера діяльності, спрямована на створення інформаційно-комунікаційних технологій (ІКТ) на всіх рівнях (від регіонального до корпоративного), національної інформаційної інфраструктури, інформаційного суспільства на основі розвитку, інтеграції та розробки інформаційних матеріалів, ресурсів комп'ютерних і телекомунікаційних засобів. У вирішенні цих завдань ключовим є питання стандартизації ІТ на основі реалізації методів і засобів архітектурно-функціональної стандартизації, що дозволяє за допомогою загальних стандартів та профілів виділити групи базових і робочих стандартів, вимог, наборів необхідних функцій і параметри впровадження конкретних ІТ-систем/інформаційних (ІС) та предметна спрямованість у сферах діяльності.

Стандарт – документ, що встановлює для загального і багаторазового застосування правила, загальні принципи або характеристики, які стосуються діяльності чи її результатів, з метою досягнення оптимального ступеня впорядкованості у певній галузі, розроблений у встановленому порядку. Міжнародний та регіональний стандарти − стандарти, прийняті міжнародним та регіональним органами стандартизації, відповідно [9, 10].

Стандартизація – діяльність, яка полягає у встановленні положень для загального і багаторазового застосування для вирішення потенційних завдань з метою досягнення оптимального ступеня впорядкування у певній сфері, результатом якої є підвищення ступеня відповідності продукції, процесів та послуг їх функціональному призначенню і сприянню науково-технічному співробітництву [9, 10].

Об'єкт стандартизації – предмети, продукція, процеси, технології, обладнання, системи, а також правила, поняття, визначення, процедури, методи тощо [9, 10].

Мета стандартизації – досягнення оптимального ступеня упорядкування в тій чи іншій галузі завдяки широкому та багатократному використанню встановлених положень, вимог, норм для вирішення реально існуючих, планованих або потенціальних задач [9, 10].

Основними результатами діяльності зі стандартизації мають бути: підвищення ступеня відповідності продукту (послуги), процесів їх функці-

онального призначення; ліквідація технічних перешкод в міжнародному товарообізі; сприяння науково-технічному прогресу в різних галузях.

Важливість принципу OSI-*взаємозв'язку відкритих систем* стала зрозумілою, коли глобалізація економіки та бізнесу в рамках Єдиного економічного простору Європи привела до необхідності стандартизації прикладних інформаційних систем і технологій.

Спочатку кожна країна чи компанія розробляла власні програмні та мережеві концепції та технічні засоби, які часто виявлялися несумісними. Різні концептуальні напрямки мали власні системи форматів даних та обміну даними, наприклад, система SWIFT у банківській справі, EDIFAST у торгівлі, промисловості та транспорті. Через відмінності в протоколах передачі та прийому даних системи часто були несумісними і не могли бути об'єднані в одне ціле. Такі ситуації сприяли розвитку міжнародної стандартизації у сфері інформаційних технологій.

Важливу роль у розвитку галузі інформаційних технологій відіграють глобальні концепції. «Відкрита система» (Open System) та «Глобальна інформаційна інфраструктура» (Global Information Infractructure), які для практичної реалізації потребують не лише розвиненої науково-методичної бази та комплексної системи стандартів, а й можуть розглядатися як найважливіші концепції. Їх метою є комплексна стандартизація масштабів ІТ.

Зусилля в науковій постановці та розробці проблем стандартизації ІТ у глобальному масштабі забезпечили розвиток відповідної системи знань і стандартів до такого рівня, що вона стає основним носієм науково-методичних засад у сфері ІТ. Ця система знань отримала назву *ітологія*. В основу розвитку *ітології* покладено такі методи [11]:

- створення основ наукового знання у вигляді методологічного ядра (метанауки), що являє собою завершену систему еталонних моделей найважливіших галузей інформатики, що структурує наукове знання загалом. Цей метод називається архітектурною специфікацією;
- розробка специфікацій для поведінки ІТ-реалізацій, тобто поведінки ІТ-систем, яку можна спостерігати на інтерфейсах (границях) цих систем. Цей метод також називають функціональною специфікацією;
- стандартизація специфікацій ІТ та управління їхнім життєвим циклом, що здійснюється системою спеціалізованих міжнародних організацій на основі строго регламентованої діяльності. Цей процес забезпечує накопичення базових атестованих наукових знань, слугує основою для створення відкритих технологій;
- розробка апарату (концепцій і методологій) для перевірки відповідності (атестації) реалізацій ІС специфікаціям ІТ, на основі яких розроблено ці ІС;
- ІТ-профілювання, тобто розробка функціональних профілів ІТ-методів з метою побудови специфікацій складних технологій шляхом поєднання базових і похідних від них (поданих у стандартизованому вигляді)

специфікацій з відповідним параметричним налаштуванням цих специфікацій (іншими словами, профілювання). Це склад оператора в просторих ІТ з основою, яка представляє базові, тобто стандартні специфікації;

- таксономія (система класифікації) ІТ-профілів, що забезпечує однозначну ідентифікацію в ІТ-просторі, прозоре подання ІТ-взаємозв'язків;
- різні методи формалізації та алгоритмізації знань, методи побудови прикладних ІТ (парадигми, мови програмування, базові відкриті технології, функціональне профілювання ІТ тощо).

В такому випадку зміст ітології потрібно розглядати передусім концептуально – як методологічну основу формалізації, аналізу та синтезу знань; по-друге, в технологічному плані – як інструмент розвитку інтелектуальних і конструктивних здібностей людини.

Таким чином отримано основні нормативно-методичні рішення. Зокрема, створено стандарти, що визначають [11]:

- глобальні концепції розвитку ІТ-галузі;
- концептуальні основи та еталонні моделі побудови основних ІТ-підрозділів;
- функції, протоколи взаємодії, інтерфейси та інші аспекти ІТ;
- мови програмування, мови специфікації ІТ, мови управління базами даних;
- моделі технологічних процесів створення та використання ІТ-систем, а також мови опису таких моделей;
- методи перевірки відповідності ІТ-систем оригінальним стандартам і профілям;
- методи та процедури функціонування поточної системи стандартів ІТ;
- метамови та нотації для опису ІТ стандартів;
- загальносистемні ІТ-функції, такі як безпека, адміністрування, інтернаціоналізація, якість обслуговування і т. д.

Стан та розвиток ІТ-стандартів нині характеризується низкою предметних галузей, які визначають поле діяльності у сфері міжнародної стандартизації. Міжнародні та національні стандарти в галузі інформатики та розробки програмного забезпечення не повністю та нерівномірно відповідають потребам у стандартизації засобів та процесів створення та застосування складних ІС. Тривалі періоди розробки, узгодження та затвердження міжнародних та національних стандартів (3–5 років) призводять до їх консерватизму та хронічного відставання від сучасних технологій створення складних ШІ. Набори стандартів для розробки сучасних інтегральних схем (профілів ІС) мають враховувати необхідність побудови інтегральних схем як відкритих систем, забезпечити їх розширюваність у разі розширення або зміни реалізованих функцій (переносимість програмного забезпечення та здатність до взаємодії з іншими інтегральними схемами). У сфері ШІ функціональні стандарти підтримують і регулюють лише найпростіші об’єкти

та рутинні, масові процеси (передача даних у мережах, програмування, документування програм і даних). Найскладніші процеси створення та розвитку великих розподілених ІС (системний аналіз і проектування, інтеграція компонентів і систем, тестування та сертифікація ІС тощо) майже не підтримують вимоги та рекомендації стандартів через різноманіття змісту, креативності характеру робіт, труднощі їх оформлення та уніфікації. Існуючі затримки з підготовкою та випуском стандартів високого рангу, а також поточна потреба стандартизації та регламентації сучасних засобів і процесів у сфері ІТ приводять до створення численних нормативних та методичних документів на рівні галузі, відділу чи компанії.

Проте розумний і послідовний відбір, удосконалення та узгодження нормативно-методичних документів у багатьох випадках дозволяє створювати на їх основі національні та міжнародні стандарти, що частково знімає проблему реалізації відкритості програмного забезпечення та ІТ-систем.

Під час визначення *середовища відкритої системи* (OSE – Open System Environment) потрібно зазначити, що основою такого середовища є розроблені, доступні та загальновизнані стандарти. Понад 300 організацій беруть участь у розробці ІТ-стандартів і специфікацій у всьому світі, і їх можна розділити на три категорії: акредитовані органи стандартизації, виробники та групи користувачів. У межах кожної з цих категорій організації об'єднуються в різні типи асоціацій, консорціумів і робочих груп (Workshops) [11].

Організаційна структура, що підтримує процес стандартизації ІТ, містить три основні групи організацій: міжнародні організації стандартизації, що входять до структури ООН, промислові професійні або адміністративні організації, промислові консорціуми [11].

Міжнародними організаціями стандартизації, що входять до структури ООН, є:

- International Organization for Standardization – ISO – Міжнародна організація зі стандартизації. Серія стандартів ISO;
- International Electrotechnical Commission – IEC – Міжнародна електротехнічна комісія. Серія стандартів ISO;
- International Telecommunication Union-Telecommunications – ITU – Міжнародний союз з телекомунікацій. До 1993 р. ця організація мала іншу назву – International Telegraph and Telephone Consultative Committee – ITTCC – Міжнародний консультативний комітет з телефонії та телеграфу, скорочено МККТТ. Серія стандартів X200, X.400, X.500, X.600.

Промислові професійні або адміністративні організації містять [11]:

- Institute of Electrical and Electronics Engineers – IEEE – Інститут інженерів з електротехніки та електроніки, міжнародна організація – творець низки важливих міжнародних стандартів у сфері ІТ). Стандарти локальної мережі IEEE802, POSIX та ін.;
- Internet Business Board – IAB. Стандарти протоколу TCP/IP;

- Open Systems Workshops – WOS – Регіональні робочі групи з відкритими системами). Профілі OSE.

Промислові консорціуми [11]:

- European Computer Manufacturers Association – ESMA – Європейська асоціація виробників обчислювальної техніки,
- Office Document Architecture – OSI – архітектура офісних документів (ODE);
- Object Management Group – OMG − група управління об'єктами;
- Common Object Request Broker Architecture – RM: загальна архітектура посередника запитів об’єктів (CORBA);
- X/Open – організований групою постачальників комп’ютерного обладнання, X/Open Portability Guide (XPG4) – Загальне прикладне середовище;
- Network Management Forum – NMF – форум з управління мережами;
- Open Software Foundation – OSF – Фонд відкритого ПЗ. Він має такі пропозиції: OSF/1 (відповідає стандартам POSIX і XPG4), MOTIF – графічний інтерфейс користувача, DCE (Distributed Computer Environment) – технологія інтеграції платформ: DEC, HP, SUN, MIT, Siemens, Microsoft, Transarc, DME (Distributed Management Environment) – технології розподіленого середовища управління.

У цій діяльності беруть участь також спеціалізовані професійні організації в різних країнах: CEN (Європейський комітет стандартизації широкого спектру товарів, послуг і технологій, зокрема пов'язаних зі сферою розробки IT, аналог ISO), CENELEC (Європейський комітет стандартизації рішень в електротехніці, зокрема стандартизації комунікаційних кабелів, волоконної оптики і електронних приладів − аналог IEC), ETSI (Європейський інститут стандартизації в галузі мережевої інфраструктури − аналог ITU-T), OMG (група об'єктно-орієнтованого управління − найбільший міжнародний консорціум, який здійснює розробку стандартів для створення уніфікованого розподіленого об'єктного програмного забезпечення, до складу якого входять понад 600 компаній − виробників програмного продукту, розробників прикладних систем і кінцевих користувачів), ECMA (Європейська асоціація виробників обчислювальних машин − міжнародна асоціація, метою якої є промислова стандартизація інформаційних і комунікаційних систем) [11].

У цій діяльності також беруть участь спеціалізовані професійні організації в різних країнах: CEN (Європейський комітет стандартизації широкого спектру товарів, послуг і технологій, зокрема пов’язаних зі сферою розвитку IT, еквівалент ISO), CENELEC (Європейський комітет стандартизації рішень в електротехніці, зокрема кабельної стандартизації зв'язку, оптичних волокон та електронних пристроїв – аналог IEC), ETSI (Європейський інститут стандартизації в сфері мережевої інфраструктури – еквівалент ITU-T), OMG (команда об'єктно-орієнтованого управління – найбільший міжна-

родний консорціум, що розробляє стандарти для створення єдиного розподіленого об'єктно-орієнтованого програмного забезпечення, містить понад 600 компаній-виробників програмного забезпечення, розробників прикладних систем і кінцевих користувачів, ECMA (European Association of Chewing Machine Computer Manufacturers – міжнародна асоціація, метою якої є промислові стандартизація інформаційно-комунікаційних систем) [11].

На рис. 1.2 подано систему міжнародних організацій, які відіграють значну роль у вирішенні завдань стандартизації ІТ.

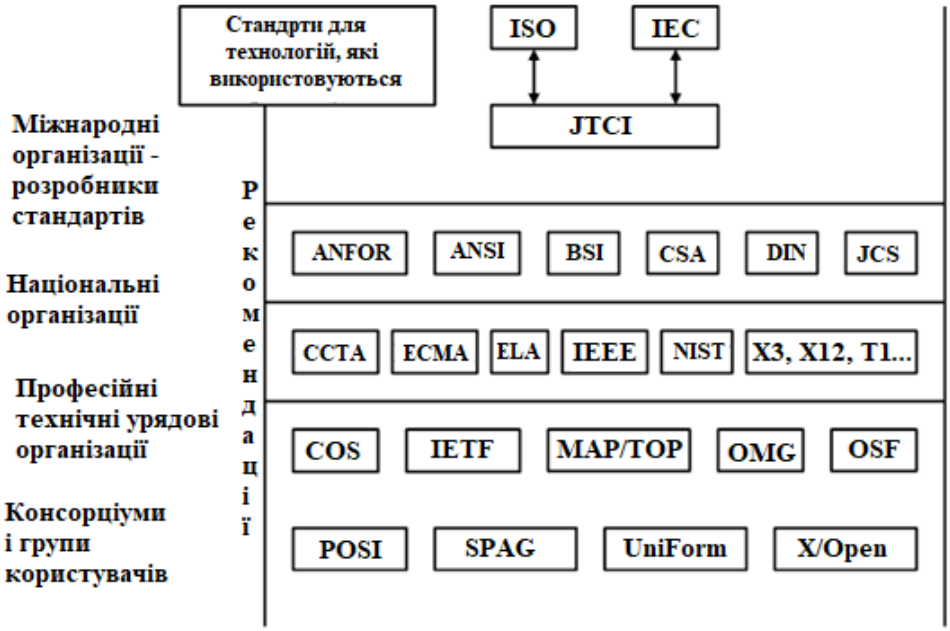


Рисунок 1.2 − Міжнародні організації й консорціуми, що є розробниками стандартів

Стандарти ISO та IEC об'єднали свою діяльність у сфері ІТ-стандартизації, щоб створити один орган JTC1 (Joint Technical Committee 1) – Об'єднаний технічний комітет № 1, спрямований на створення комплексної системи основних ІТ-стандартів та їх розширення на конкретні сфери діяльність.

Основні цілі комітету JTC1 – це розробка, підтримка та просування ІТ-стандартів, необхідних на світовому ринку, що відповідають вимогам бізнесу та користувачів і важливі [11]:

- проектування та розробка ІТ-систем та інструментів;
- продуктивність і якість ІТ-продуктів і систем;
- безпека ІТ-систем та інформації;
- передані прикладні програми;
- сумісність продуктів та ІТ-систем;
- гармонізовані заходи та середовище;
- гармонізований глосарій ІТ термінів;

- зручний і ергономічний інтерфейс користувача.

На рис. 1.3 подано загальну схему стандартизації ІТ.

Рисунок 1.3 − Схема функціональної стандартизації ІТ

Робота над стандартами ІТ в JTC1 тематично розділена на підкомітети (Subcommittees – SC), пов'язані з розробкою ІТ-стандартів, що належать до середовища відкритих систем OSE.

Нижче наведено назви деяких таких комітетів і підкомітетів [11]:

- С2 − символьні набори та кодування інформації;
- С2 – набори символів і кодування інформації;
- SC6 – телекомунікації та обмін інформацією між системами;
- SC7 – розробка програмного забезпечення та системна документація;
- SC18 – текстові й офісні системи;
- SC21 – відкрита розподілена обробка (ODP – Open Distributed Processing), керування даними (DM – Data Management) і взаємозв’язок відкритих систем OSI;
- SC22 – мови програмування, їх середовища та системні програмні інтерфейси;
- SC24 – комп’ютерна графіка;
- SC27 – загальні методи безпеки ІТ-додатків;
- SGFS – це спеціальна група, яка займається функціональними стандартами.

Результатом цілеспрямованої діяльності зі стандартизації стало створення розвиненої системи стандартів, що охоплює весь спектр основних

напрямів ІТ: глобальні концепції розвитку ІТ-галузі; основні еталонні моделі; методичні вказівки; специфікації типових аспектів розробки, тестування, експлуатації та використання ІТ-систем.

Наразі в усьому світі існує кілька авторитетних спільнот розробки стандартів відкритих систем. Проте найважливішою діяльністю в цій галузі є робота IEEE над робочими групами та комітетами з інтерфейсу портативних операційних систем (Portable Operating System Interface – POSIX). Перша робоча група POSIX була створена в IEEE в 1985 році на основі комітету стандартів UNIX (зараз UniForum). Звідси вихідна орієнтація роботи POSIX на стандартизацію інтерфейсів операційної системи UNIX. Однак поступово обсяг роботи робочих груп POSIX розширився до такої міри, що можна було говорити не тільки про стандартну операційну систему UNIX, а й про POSIX-сумісні операційні середовища, тобто про будь-які операційні середовища, інтерфейси яких відповідають POSIX технічні умови [11].

Міжнародні стандарти мають бути впроваджені для кожного системного компонента мережі, включно й кожна операційна система та пакети програм. Поки компоненти відповідають таким стандартам, вони відповідають цілям відкритих систем. Характерною особливістю сучасних міжнародних ІТ-стандартів є те, що вони містять означення основних понять і термінів у сфері ІТ, описи моделей, сценаріїв, функцій, правил поведінки та подання інформації. Іншими словами, властивості ІТ/ІС подано в стандартах у вигляді концептуальних, функціональних та інформаційних моделей об’єктів стандартизації.

Питання для самоконтролю

1. Розкрийте суть термінів «парадигма» та «зміна парадигм».
2. Охарактеризуйте інформаційно-технологічна парадигму та наведіть її характеристики.
3. Охарактеризуйте зміну парадигм ІТ в контексті теоретико-методологічних основ наукового пошуку.
4. Наведіть приклади змін парадигм в науці та в інформаційних технологіях.
5. Розкрийте суть термінів «стандарт», «стандартизація», «об’єкт стандартизації».
6. Охарактеризуйте принцип взаємозв'язку відкритих систем OSI та яке його значення.
7. Охарактеризуйте науково-методичні основи і стандарти в сфері інформаційних технологій.
8. Розкрийте суть терміна «ітологія» та охарактеризуйте методи, які є основою розвитку ітології.
9. Охарактеризуйте стан і розвиток стандартів ІТ.
10. Охарактеризуйте організаційну структуру, що підтримує процес стандартизації ІТ.

2 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЕНЬ

2.1 Організація паралельних обчислень з використанням сучасних технологій

Останні десятиліття високопродуктивні обчислювальні системи знаходять своє застосування в процесі вирішення практично будь-яких завдань науки і техніки в усіх галузях народного господарства. Серед таких завдань – моделювання різних фізичних процесів, задачі обчислювальної хімії та біології, нанотехнології, автоматизація проектування та багато інших. Прогрес у галузі високопродуктивних обчислень багато в чому визначає темп розвитку науки і техніки, і, в остаточному підсумку, рівень технологічного розвитку держави в цілому. Тому, можна з упевненістю стверджувати, що створення і вивчення методів розробки програмно-апаратного забезпечення для високопродуктивних обчислювальних систем є однією із найважливіших задач сучасних інформаційних технологій [12].

Паралельні обчислення є узагальненим терміном, що застосовується для позначення технологій та методів розробки програмно-апаратного забезпечення для високопродуктивних комп'ютерних систем. Тому термін «паралельні обчислення» описує достатньо широку галузь, яка пов'язана з організацією розрахунків в обчислювальних системах, що містять декілька процесорних пристроїв. До таких систем відносять багатоядерні процесори, багатопроцесорні системи із загальною пам'яттю, високопродуктивні обчислювальні кластери з розподіленою пам'яттю або гібридною архітектурою, системи, що реалізують загальні обчислення на основі відеоадаптерів (GPGPU), хмарні обчислення (Cloud Computing) тощо.

У наукових публікаціях, дослідженнях і прикладних проектах паралельним обчисленням останнім часом приділяється велика увага. Це пов'язано, переважно, з двома чинниками. Перший фактор обумовлений науково-технічним прогресом, внаслідок якого з'явилися нові галузі знань, що потребують застосування високопродуктивних методів математичного моделювання. Відповідно і моделі також істотно ускладнились. У підсумку – спостерігається тенденція зростання потреби в ресурсоємних розрахунках, які в ряді випадків можна виконати лише на базі високопродуктивної обчислювальної техніки і виключно за допомогою методів паралельних, розподілених або ж гетерогенних обчислень.

Другий істотний фактор, внаслідок якого інтерес до паралельних обчислень суттєво зріс, полягає в значному поширенні паралельних комп'ютерів. Останнім часом багатопроцесорні сервери можна часто зустріти на середніх і великих підприємствах, у банках, дослідних інститутах та обчислювальних центрах. З появою багатоядерних

процесорів та відеоадаптерів багато користувачів стали володарями своєрідних «міні-суперкомп'ютерів» на своїх робочих місцях. Істотний прогрес у галузі мережевих технологій дозволив використовувати для паралельних обчислень локальні мережі підприємств, навчальні класи освітніх установ, уможливив створення відносно недорогих обчислювальних кластерів.

Можна стверджувати, що «паралельні інформаційні технології» перетворилися з вузькоспеціальної дисципліни в необхідну складову інформаційної інфраструктури різноманітних установ та організацій – з одного боку, а з другого боку – комплексу знань фахівців з інформаційних технологій та комп'ютерної інженерії, розробників сучасного програмно-апаратного забезпечення.

Зокрема, актуальним є застосування паралельних обчислень в галузях, пов'язаних з проведенням ресурсоємних та складних розрахунків, а саме у [12]:

– ***системах підтримки проектування (CAD – Computer Aided Design).*** У таких системах необхідність здійснювати моделювання в реальному часі висуває високі вимоги до продуктивності програмно-апаратного забезпечення. Внаслідок застосування паралельних інформаційних технологій вдається прискорити процес проектування, і тим самим, знизити часові та трудові витрати на розробку нової моделі;

– ***складних інженерних розрахунках та імітаційному моделюванні.*** До цього класу належать різноманітні задачі з галузі моделювання аварійних ситуацій, моделювання робастних систем, міцнісного моделювання і багато інших;

– ***математичному моделюванні фізичних процесів.*** До цього широкого класу задач відносять сфери дослідження динаміки рідини і газу, електромагнітні і ядерні взаємодії, процеси горіння тощо. Такі процеси, як правило, описуються системами рівнянь в часткових похідних. Застосування для вирішення таких задач різницевих методів найчастіше потребує дуже великих обсягів обчислень і пам'яті. Використання багатопроцесорних систем і методів паралельних та розподілених обчислень дозволяє підвищити показники точності та продуктивності моделювання;

– ***моделюванні глобальних процесів.*** Насамперед, це задачі прогнозування зміни клімату, природних катаклізмів тощо. Також великою обчислювальною складністю характеризуються різноманітні геологічні дослідження, пов'язані з аналізом будови та процесів у надрах планети;

– ***обчислювальній хімії.*** Різноманітні задачі цієї галузі спрямовані на вивчення властивостей речовини в різних станах. Широке застосування методів молекулярної динаміки також часто потребує значних обчислювальних ресурсів, що підтверджує актуальність застосування паралельного програмування. До цієї категорії можна також віднести задачі, пов'язані з

оптимальною конфігурацією протеїнів, розшифрування ДНК і багато інших проблем, суміжних з хімічною галуззю;

– ***бізнес-додатках.*** До цієї категорії відносяться задачі, пов'язані з аналізом фінансових ринків і прогнозуванням курсів валют. Також поширені оптимізаційні задачі для формування прогнозу та прийняття рішення щодо найкращого варіанта використання фінансових або інших ресурсів, побудови оптимальних транспортних і телекомунікаційних мереж, розміщення підприємств в регіоні тощо.

Цей перелік можна продовжувати й далі, адже він є досить великим. Проте, вище наведено лише деякі, найбільш актуальні із численних застосувань систем паралельних обчислень, перелік сфер застосування яких неухильно і активно розширюється [12].

2.2 Класифікація сучасних паралельних обчислювальних систем

Навіть короткий перелік типів сучасних паралельних обчислювальних систем (ОС) дає зрозуміти, що для орієнтування в цьому різноманітті необхідна чітка система класифікації. Серед усіх розглянутих систем класифікації ОС найбільше визнання отримала класифікація, запропонована в 1966 році М. Флінном. В її основу покладено поняття потоку, під яким розуміється послідовність елементів, команд або даних, яка обробляється процесором. Залежно від кількості потоків команд і потоків даних Флінн виділяє чотири класи архітектур: SISD, MISD, SIMD, MIMD (табл. 2.1, рис. 2.1) [13].

Таблиця 2.1 – Класифікація архітектур комп’ютерних систем

Потік команд	**Одиничний потік даних (ОД)**	**Множинний потік даних (МД)**
Одиничний (ОК)	ОКОД (SISD) (однопроцесорні комп’ютери)	ОКМД (SIMD) (комп’ютери з паралельними або асоціативними процесорами)
Множинний (МК)	МКОД (MISD) (конвеєрні магістральні комп’ютери)	МКМД (MIMD) (багатопроцесорні або багатомашинні комплекси)

SISD (Single Instruction Stream/Single Data Stream) − одиничний потік команд і одиничний потік даних. Представник цього класу – класичний фон-нейманівський комп'ютер. Команди обробляються послідовно і кожна команда ініціює одну операцію з одним потоком даних. До цього класу комп'ютерів можна віднести також конвеєрні комп'ютери. Деякі спеціалісти вважають, щодо SISD-систем можна віднести і векторно-конвеєрні ОС, якщо розглядати вектор як неподільний елемент даних для відповідної команди.

MISD (Multiple Instruction Stream/Single Data Stream) – множинний потік команд і одиничний потік даних. В архітектурі присутня множина процесорів, які обробляють один і той самий потік даних. Ряд дослідників відносять до цього класу конвеєрні системи. Прийнято вважати, що доки цей клас незадіяний, він може бути корисним для розробки принципово нових концепцій побудови обчислювальних систем.

SIMD (Single Instruction Stream/Multiple Data Stream) – одиничний потік команд і множинний потік даних. Ця архітектура дозволяє виконати одну арифметичну операцію відразу над багатьма даними – елементами вектора. Представниками цього класу є системи з матрицею процесорів, де один керівний пристрій контролює множину процесорних елементів. Усі процесорні елементи отримують від пристрою управління однакову команду і виконують її над власними локальними даними. В цей клас можуть бути віднесені і векторно-конвеєрні ОС, якщо кожний елемент вектора розглядати як окремий елемент даних [13].

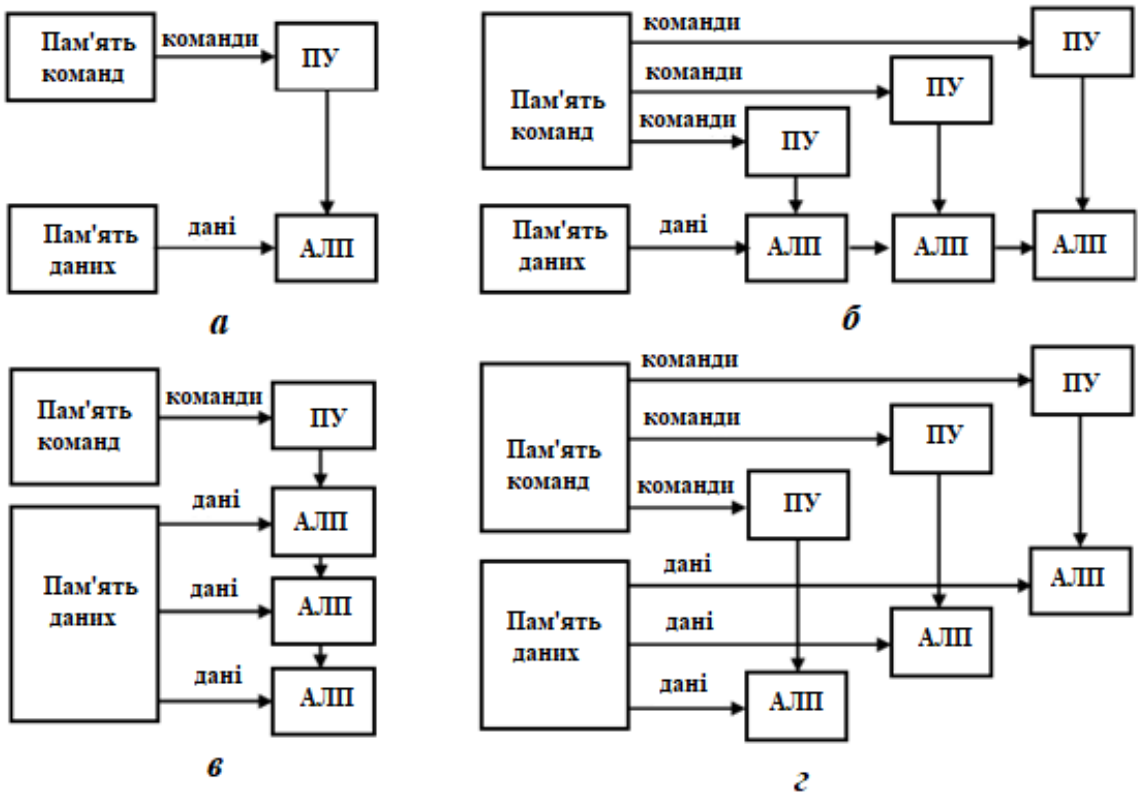


Рисунок 2.1 – Архітектура обчислювальних систем за Флінном: *а* – SISD; *б* –MISD; *в* – SIMD; *г* – MIMD

MIMD (Multiple Instruction Stream/Multiple Data Stream) – множинний потік команд і множинний потік даних. До цього класу відносяться системи з множиною пристроїв обробки команд, які об'єднані в єдиний комплекс і кожний працює з власним потоком команд. Цей клас надзвичайно широкий, оскільки містить в собі різного роду мультипроцесорні системи.

Запропонована схема класифікації аж до теперішнього часу є однією із найпопулярніших за початковою характеристикою певної обчислювальної системи. Наприклад, якщо відзначено, що комп'ютер належить до класу ОКМД (SIMD) або МКМД (MIMD), то відразу стає зрозумілим базовий принцип його роботи, і в деяких випадках цього буває достатньо.

Однак є явні **недоліки**. Зокрема, класифікація Флінна надто загальна, наприклад, відносить усі паралельні комп'ютери, крім мультипроцесорних, до одного класу і не вказує ніякої відмінності між конвеєрними комп'ютерами і матрицею МП. Також деякі обчислювальні системи (ОС) заслуговують більшої уваги щодо архітектури, наприклад, dataflow і векторно-конвеєрні машини чітко не вписуються в класифікацію Флінна.

Інший недолік – це надмірна заповненість класу МКМД (MIMD). Зважаючи на викладене, відзначимо, що є потреба у таксономіях, які більш вибірково систематизують архітектури, що за таксономією Флінна потрапляють в один клас, але зовсім різні за кількістю процесорів, природою і топологією зв'язків між ними, за способом організації пам'яті і, звичайно ж, за технологією програмування, зокрема паралельного програмування [13].

Розширення класифікації М. Флінна із врахуванням паралельних обчислювальних систем

Необхідно відзначити, що з розвитком паралельних обчислень та відповідних комп'ютерних технологій була розширена відома класифікація Майкла Флінна (M. J. Flynn). Схеми SISD, SIMD, MISD, MIMD були розширені для паралельних обчислень до SPMD (Single-Program / Multiple-Data – одна програма, кілька потоків даних) і MPMD (Multiple-Programs / Multiple-Data – множина програм, велика кількість потоків даних), відповідно [12].

Схема SPMD (SIMD) дозволяє декільком процесорам виконувати одну і ту саму інструкцію або програму за умови, що кожен процесор отримує доступ до різних даних.

Схема MPMD (MIMD) дозволяє працювати декільком процесорам, причому всі вони виконують різні програми або інструкції і користуються власними даними.

Таким чином, в одній схемі всі процесори виконують одну і ту саму програму або інструкцію, а в іншій всі процесори виконують різні програми або інструкції. Звичайно ж, можливі гібридні реалізації цих моделей, в яких процесори можуть бути розділені на групи, з яких одні утворюють SPMD-модель, а інші – MPMD-модель. За використання схеми SPMD всі процесори просто виконують одні й ті самі операції, але з різними даними. Якщо ж застосовується схема MPMD, всі процесори виконують різні види робіт, і хоча в цьому випадку всі вони разом намагаються вирішити одну задачу, кожному з них визначається свій аспект цієї задачі.

Сучасна класифікація паралельних обчислювальних систем

Розглянемо класифікацію паралельних ОС з урахуванням новоявлених сучасних архітектур. В основу класифікації (рис. 2.2) покладемо чотири базових класи класифікації М. Флінна (SISD, SIMD, MISD, MIMD), які розбиваються на підкласи згідно з доповненнями учених Ванга та Бріггса [12].

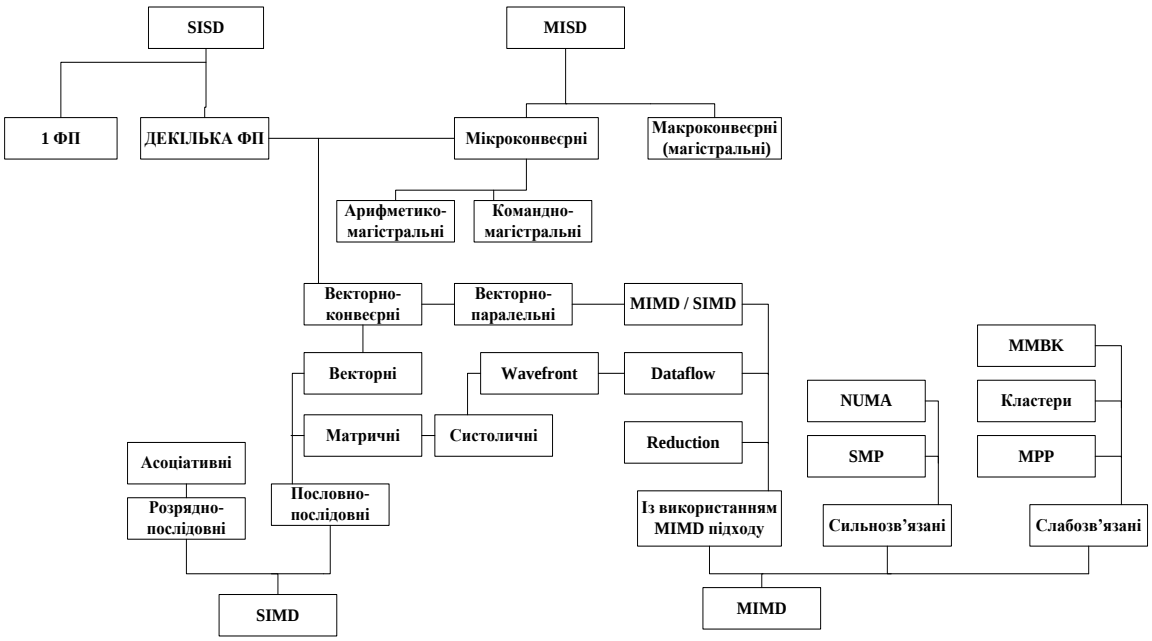


Рисунок 2.2 – Класифікація обчислювальних систем

Клас SISD поділяється на системи з одним функціональним пристроєм (ФП) і декількома ФП. Клас SIMD утворюють два підкласи – розрядно-послідовних і послівно-послідовних ОС. У класі MIMD виділимо сильно- і слабкозв'язані ОС, а також ті ОС, що використовують ідеї MIMD, підкласи яких (MIMD/SIMD, Dataflow, Reduction, Wavefront) утворюються відповідно до класифікації Дункана. Нарешті, до класу MISD віднесемо комп'ютерні засоби та ОС, що використовують ідею конвеєрного оброблення. Паралелізм на рівні операторів і команд реалізується в мікроконвеєрних системах. Водночас системи, здатні розділяти за рівнями безпосереднє виконання однієї команди, утворюють підклас арифметико-магістральних систем. А системи, які конвеєризують всі етапи виконання команди (вибірка з пам'яті, дешифрація, вибірка операндів, виконання, запис результатів), визначають як командно-магістральні. Крім цього, виділено підклас макроконвеєрних (магістральних) ОС, здатних багаторазово вирішувати одну і ту саму обчислювальну задачу. В такому разі задача розбивається на послідовні частини, кожна з яких виконується на окремому процесорі (пристрої обробки). Вихідні дані однієї частини є вхідними – для наступної. За рахунок організації подібного макроконвеєра за багаторазового повторення обчислень з різними вхідними даними отримують істотний виграш у продуктивності.

Переходячи до підкласів SIMD, помітно, що до розрядно-послідовних SIMD-систем можна віднести асоціативні ОС, а до послівно-послідовних – векторні та матричні. Систолічні масиви можна розглядати як окремий підклас матричних архітектур.

Серед сильнозв'язаних MIMD-архітектур виділяють симетричні мультипроцесори (SMP), що мають зосереджену загальну пам'ять, і архітектури з неоднорідним доступом до пам'яті (NUMA), в яких логічна спільна пам'ять фізично розподілена по вузлах системи. Слабкозв'язані MIMD-системи подано масивно-паралельними (MPP), кластерними архітектурами, а також багатомашинними обчислювальними комплексами (БМОК). MPP-системи являють собою сукупність спеціалізованих обчислювальних модулів, об'єднаних високошвидкісними міжпроцесорними каналами зв'язку. Кластерні системи також являють собою приклад масового паралелізму, проте їх побудова здійснюється на основі стандартних промислових комплектуючих.

На рис. 2.2 також показано, що ряд ОС можуть бути одночасно віднесені до декількох класів або підкласів. Поєднання принципів мікроконвеєрного і векторного оброблення дає векторно-конвеєрні архітектури. Привносячи в них ідеї MIMD/SIMD, отримуємо векторно-паралельні (PVP) ОС. Машини *wavefront* є гібридними ОС, що керовані потоком даних (*dataflow*) і систолічними масивами [12].

Використовуються й інші класифікації архітектур, зокрема систематика Ф. Шара, структурна систематика Р. Хокні та К. Джессхоупа [13].

Структурна систематика Р. Хокні та К. Джессхоупа

На першому рівні всі обчислювальні системи поділяються за принципом множинності (кількості) на однокомп'ютерні та багатокомп'ютерні. Обчислювальні системи з одним комп'ютером, зі свого боку, поділяються на ОМ з одним конвеєрним МП та з багатьма МП.

Перші з них є традиційними послідовними комп'ютерами, а другі утворюють клас паралельних комп'ютерів, які поділяються на конвеєрні, неконвеєрні та мікропроцесорні матриці.

Прикладом однієї з перших неконвеєрних обчислювальних машин з паралелізмом є комп'ютер CDC-6600, побудований на основі декількох скалярних процесорів.

Конвеєрні ЕОМ поділяються на такі, що виконують тільки скалярні команди, наприклад, комп'ютери CDC-7800, FPC AP-120B, і такі, що виконують векторні команди. Комп'ютери, що використовують векторні команди, поділяють, також, на комп'ютери зі спеціалізованим конвеєром, наприклад CRAY-1, та з універсальним конвеєром – комп'ютер CYBER 205.

Комп'ютери з класу машин з матрицею процесорів поділяють за зв'язаністю процесорів у матриці, розрядністю тощо. Першими машинами такого типу були ILLIAC-IV, BSP, STA-RAN, ICL DAP, OMEN та ін [13].

2.3 Рівні паралелізму

В умовах постійно зростаючих вимог до продуктивності обчислювальної техніки все більш явними стають обмеження класичної фон-нейманівської архітектури. Подальший розвиток обчислювальної техніки пов'язаний з переходом до паралельних обчислень як у рамках однієї обчислювальної машини, так і шляхом створення багатопроцесорних систем і мереж, які об'єднують велику кількість окремих процесорів або окремих обчислювальних машин. Для такого підходу замість терміна «обчислювальна машина» більш підходить термін «обчислювальна система» (ОС). Головною особливістю такої системи є наявність у ній засобів, які реалізують паралельну обробку [13].

Паралельна обробка інформації являє собою одночасне рішення двох або більшої кількості частин однієї й тієї самої програми двома чи більшою кількістю ЕОМ (процесорними елементами) обчислювальної системи. Реалізують три основних способи організації паралельної обробки [13]:

1) суміщення у часі різноманітних етапів різних задач – це мультипрограмна обробка, яка широко використовується як у однопроцесорних ЕОМ, так і в складних ОС;

2) одночасне розв'язання різноманітних задач або частин однієї задачі (можливо тільки за наявності декількох обробних пристроїв);

3) конвеєрна обробка інформації;

Перші два способи використовують особливості паралельних задач або потоків задач, що дозволяє здійснювати той або інший паралелізм. Перший тип паралелізму – це *природний паралелізм незалежних задач*, який полягає у тому, що в систему надходить безперервний потік непов'язаних між собою задач. У цьому випадку розв'язання будь-якої задачі не залежить від результатів розв'язання інших задач, що дозволяє підвищити продуктивність ОС у разі використання декількох обробних пристроїв.

Одним з найпоширеніших типів паралелізму є *паралелізм незалежних гілок*. Суть його полягає у виділенні окремих незалежних частин великої задачі (гілок програми), які можуть виконуватись паралельно окремими обробними пристроями незалежно один від одного. Причому обробні пристрої ОС функціонують в однопрограмному режимі паралельної обробки інформації. Двома незалежними гілками програми визнаються гілки програми, що відповідають таким умовам [13]:

- ні одна з вхідних для гілки програми величин не є вихідною величиною іншої програми (відсутність функціональних зв'язків);
- для двох гілок програми не має здійснюватись запис у одні й ті самі комірки пам'яті (відсутність зв'язку за оперативною пам'яттю);
- умови виконання однієї гілки не залежать від результатів, що отримані під час виконання іншої гілки (незалежність за управлінням);

• обидві гілки мають виконуватись у різних блоках програми (програмна незалежність).

Виділення незалежних гілок широко використовується під час паралельної обробки в задачах матричної алгебри, лінійного програмування, спектральної обробки сигналів, прямого та оберненого перетворення Фур'є та ін.

Методи та засоби реалізації паралелізму залежать від того, на якому рівні він має забезпечуватись. Зазвичай розрізняють такі *рівні паралелізму* [13]:

• ***рівень завдань –*** декілька незалежних завдань одночасно виконуються на різних процесорах, які практично не взаємодіють один з одним. Цей рівень реалізується в ОС з множиною процесорів у багатозадачному режимі;

• ***рівень програм*** – частини однієї задачі виконуються множиною процесорів. Цей рівень досягається на паралельних ОС;

• ***рівень команд*** – виконання команди розділяється на фази, а фази декількох послідовних команд можуть бути перекриті за рахунок конвеєризації. Цей рівень використовується в ОС з одним процессором;

• ***рівень бітів (арифметичний рівень)*** – біти слова обробляються один за одним. Цей процес має назву *біт-послідовна операція*. Якщо біти слова обробляються одночасно, кажуть про *біт-паралельну операцію*. Цей рівень реалізується в звичайних і суперскалярних процесорах.

Паралелізм рівня завдання можливий між незалежними завданнями або їх фазами. Основним засобом реалізації паралелізму на рівні завдань є багатопроцесорні і багатомашинні обчислювальні системи, в яких завдання розподіляються за окремими процесорами або машинами. Однак, якщо кожне завдання трактувати як сукупність незалежних задач, реалізація цього рівня можлива і в рамках однопроцесорної ОС. У цьому випадку декілька завдань можуть одночасно знаходитись в основній пам’яті ОС, за умови, що в кожний момент виконується тільки одне з них. Коли завдання, яке виконується, потребує введення/виведення, ця операція запускається, а до її завершення інші ресурси ОС передаються другому завданню. Після завершення введення/виведення ресурси ОС повертаються до завдання, яке ініціювало цю операцію. В цьому випадку паралелізм забезпечується за рахунок того, що центральний процесор і система введення/виведення функціонують одночасно і обслуговують різні завдання.

Паралелізм виникає також, коли у незалежних завдань, які виконуються в ОС, є декілька фаз, наприклад, обчислення, запис у графічний буфер, системні виклики. За те, як різні завдання впорядковуються і витрачають загальні ресурси, відповідає операційна система.

Паралелізм рівня програм. Про паралелізм на рівні програми можна говорити у двох випадках. По-перше, коли в програмі можна виділити незалежні ділянки, які допустимо виконувати паралельно. Другий тип

паралелізму програм можливий у межах окремого програмного циклу, якщо в ньому окремі ітерації не залежать одна від одної. Програмний паралелізм можна реалізувати за рахунок великої кількості процесорів або множини функціональних блоків.

Загальна форма паралелізму на рівні програм організується розбиттям даних, які програмуються на підмножини. Цей розподіл називають *декомпозицією області* (domain decomposition), а паралелізм, який виникає, має назву *паралелізму даних*. Підмножини даних призначаються різним обчислювальним процесам, і цей процес має назву *розподілення даних* (data distribution). Процесори виділяються певним процесам або за ініціативою програми, або в процесі роботи операційною системою. На кожному процесорі може виконуватись більше одного процесу.

Паралелізм рівня команд. Паралелізм на рівні команд має місце, коли обробка декількох команд або виконання різних етапів однієї і тієї самої команди може перекриватись у часі. Розробники обчислювальної техніки ще здавна зверталися до методів, відомих під загальною назвою «поєднання операцій», за якого апаратура ОМ у будь-який момент часу виконує одночасно більше однієї операції. Цей загальний принцип містить два поняття: *паралелізм* і *конвеєризація*. Хоча у них багато загального і їх часто важко розрізняти на практиці, ці терміни відображають два принципово різних підходи.

У першому варіанті поєднання операцій досягається за рахунок того, що в складі обчислювальної системи окремі пристрої присутні в декількох копіях. Так, до складу процесора може входити декілька АЛП, і висока продуктивність забезпечується за рахунок одночасної роботи всіх цих АЛП.

Під час організації паралельного обчислювального процесу виникає задача вибору побудови розкладу.

Розклад паралельних обчислювальних процесів визначає порядок виконання програми в ОС, включно й розподіл частин програми по обробних пристроях (процесорах, ОМ), і слугує основою алгоритмів планувальника операційної системи та різноманітних керівних програм.

Як критерії оптимальних розкладів для паралельної програми можна назвати [13]:

- мінімізацію часу виконання програми;
- мінімізацію кількості потрібних пристроїв обробки;
- мінімізацію середнього часу закінчення виконання завдань;
- максимізацію завантаження пристроїв ОС;
- мінімізацію часу простоювання пристроїв;
- найчастіше використовують перший критерій.

2.4 Комбінування паралельних та розподілених обчислень

Як уже зазначалось, під час паралельних обчислень відразу декілька інструкцій можуть виконуватися в один і той самий момент часу. Одна інструкція розбивається на кілька дрібних частин, які будуть виконуватися одночасно (їх можуть бути сотні чи навіть тисячі). Таким чином, *у паралельних обчисленнях послідовність і місце розташування складових програмного забезпечення не завжди передбачувані.* Декілька завдань можуть одночасно почати виконуватися на будь-якому процесорі без гарантії того, що завдання закріплені за певними процесорами, або ж що певна задача завершиться першою, або всі задачі завершаться у певному порядку.

Крім паралельного виконання задач, тут можливе паралельне виконання частин однієї задачі (підзадач). У деяких конфігураціях не виключена можливість виконання підзадач на різних процесорах або, навіть, різних КС. На рис. 2.3 зображено три рівні паралелізму, які можуть бути присутні в одній комп'ютерній програмі [12].

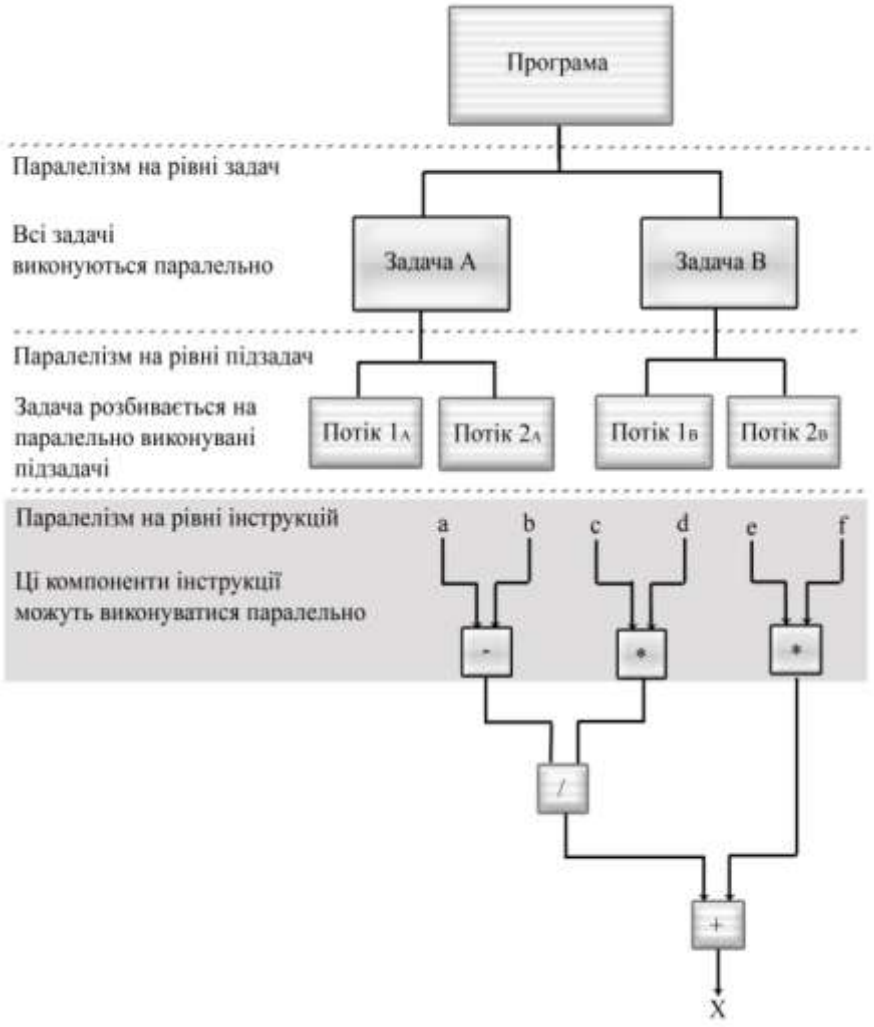
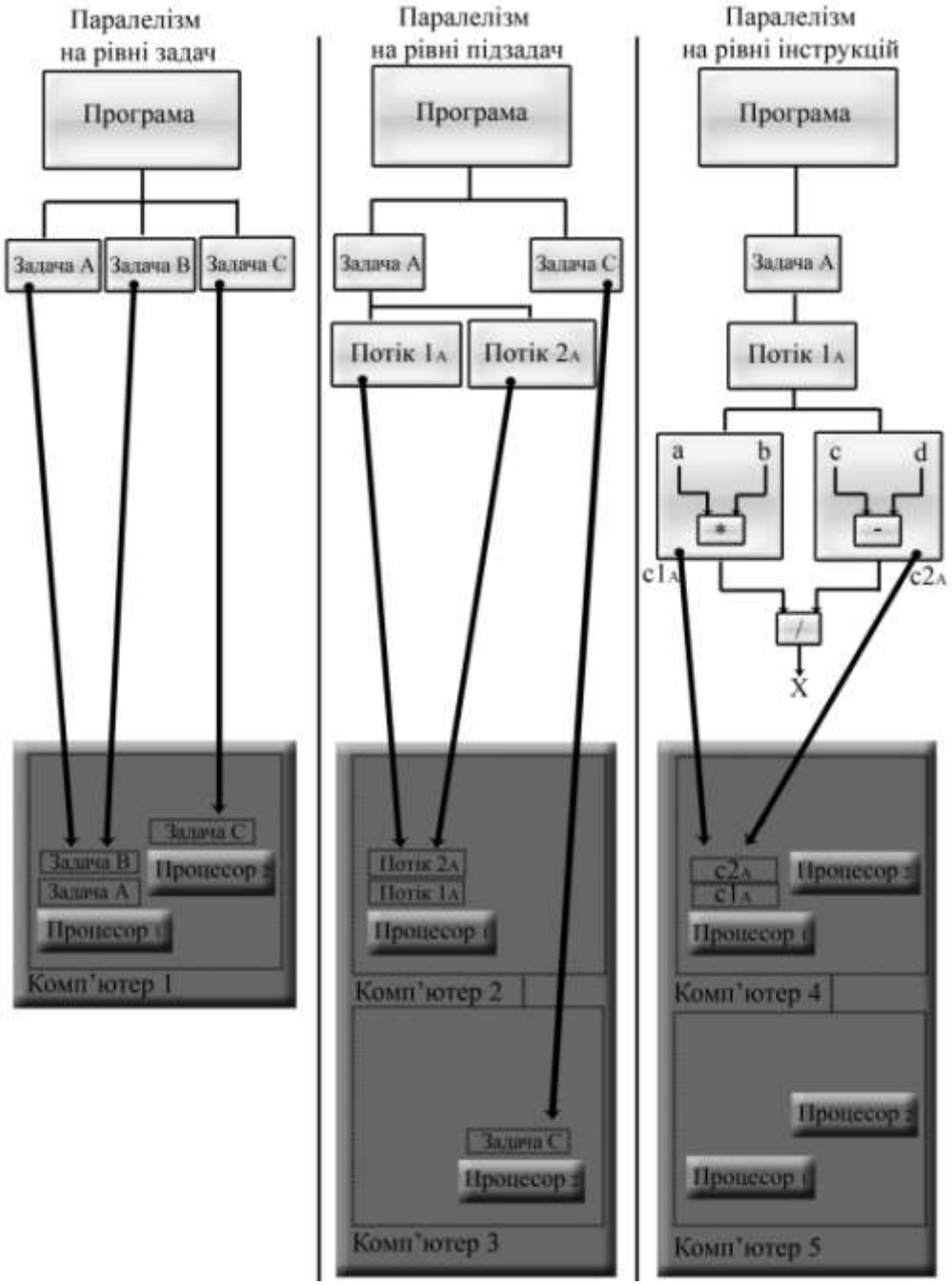


Рисунок 2.3 – Рівні паралелізму під час організації обчислювального процесу

Модель програми (рис. 2.3) відображає кардинальну зміну парадигми обчислень. Тут відображено три рівні паралелізму та їх розподіл на декілька процесорів. Поєднання цих трьох рівнів з базовими паралельними конфігураціями процесорів показано на рис. 2.4 [12].



а) б) в)

Рисунок 2.4 – Рівні паралелізму із врахуванням базових паралельних конфігурацій процесорів:
а) – однопроцесорна КС; б) – PVM-середовище з декількома однопроцесорними КС; в) – PVM-середовище з декількома багатопроцесорними КС

Додатково необхідно враховувати те, що декілька задач може виконуватися на одному процесорі, навіть за наявності в КС декількох процесорів. Така ситуація створюється системними стратегіями планування. На тривалість виконання задач, підзадач та інструкцій впливають і обрані стратегії планування, і пріоритети процесів, потоків, і швидкодія пристроїв введення-виведення.

Також потрібно враховувати різноманітність архітектур під час переходу від послідовної моделі програмування до паралельної (рис. 2.4). *Основна відмінність тут полягає в переході від строго впорядкованої послідовності задач до лише частково впорядкованої (або зовсім невпорядкованої) колекції задач.* Тобто, паралелізм перетворює раніше відомі величини (порядок виконання, час виконання і місце виконання) на невідомі. Будь-яка комбінація цих невідомих величин є причиною трансформаційних змін значень програми, причому зазвичай непередбачуваним чином.

Зважаючи на вищенаведене, необхідно відзначити, що процес проектування паралельних і розподілених КС та їх програмно-апаратного забезпечення має містити три складові: декомпозицію, зв'язок та синхронізацію [12].

Декомпозиція являє собою процес розбиття задачі та її розв’язання на частини. Іноді частини групуються в логічні області (наприклад, пошук, сортування, обчислення, введення/виведення даних і т. д.). В інших випадках частини групуються за логічними ресурсами (наприклад, файл, зв'язок, база даних і т. д.). *Декомпозиція програмного рішення часто зводиться до декомпозиції робіт – WBS (Work Breakdown Structure), яка визначає, що мають робити різні частини програмного забезпечення.* Однією з основних проблем паралельного програмування є ідентифікація природної декомпозиції робіт для програмного вирішення. Варто зазначити, що не існує простого та однозначного підходу до ідентифікації WBS. Це складний процес переведення принципів, ідей, шаблонів, правил, алгоритмів або формул в набір інструкцій, що виконуються, і даних, що обробляються КС. Це, переважно, і розкриває природну декомпозицію робіт програмного рішення. Чим краще модель зрозуміла і розроблена, тим більш природною буде декомпозиція робіт.

Після декомпозиції програмного рішення на ряд паралельно виконуваних частин зазвичай виникає питання про зв'язок цих частин між собою. На цьому етапі розглядається комплекс проблемних питань щодо: реалізації зв'язків під час розподілу підзадач/задач по різних процесах або різних КС; спільного використання загальної області пам'яті різними частинами програмного забезпечення; комунікації між КС та повідомлення про завершення виконання підзадачі/задачі; черговості виконання підзадач/задач; комунікації між КС та повідомлення про відмову виконання тощо. Якщо ж окремим частинам програмного забезпечення не

потрібно зв'язуватися між собою, отже, вони насправді не утворюють єдиний додаток.

Синхронізація актуалізується в процесі виконання декомпозиції робіт. Коли компоненти програмного забезпечення працюють у рамках однієї задачі, їх функціонування необхідно координувати. Потрібно, щоб певний компонент мав можливість визначити, коли досягається розв’язання усієї задачі. Важливо також скоординувати порядок виконання компонентів. В цьому випадку також виникає комплекс проблемних питань щодо: одночасності виконання задачі та визначення черговості і організації перебування процесорів в стані очікування; організації коректного доступу до спільних ресурсів та пріоритетів; організації подальшої роботи процесорів після виконання підзадачі/задачі або ж у випадку дострокового виконання, тощо. Таким чином, декомпозиція, зв'язок і синхронізація – це той мінімум питань, які необхідно вирішити, приступаючи до паралельного або розподіленого програмування [12].

2.5 Високопродуктивні обчислювальні системи з гібридною архітектурою

Методи та середовища організації високопродуктивних паралельних і розподілених обчислень

Message Passing Interface.

Message Passing Interface (MPI, інтерфейс передачі повідомлень) – програмний інтерфейс (API) для передачі інформації, який дозволяє обмінюватися повідомленнями між процесами, які виконують одну задачу.

MPI є найпоширенішим стандартом інтерфейсу обміну даними в паралельному програмуванні. MPI реалізований для великої кількості комп'ютерних платформ, і використовується під час розробки програм для кластерів і суперкомп'ютерів. Основним засобом комунікації між процесами в MPI є передача повідомлень один одному.

У моделі програмування, яку підтримує MPI, програма породжує кілька процесів, що взаємодіють між собою за допомогою звернень до підпрограм передачі і прийому повідомлень. Зазвичай, під час ініціалізації MPI-програми створюється фіксований набір процесів, причому кожен процес виконується на своєму процесорі. У цих процесах можуть виконуватися різні програми, тому модель програмування MPI іноді називають MPMD-моделлю (Multiple Program Multiple Data – безліч програм безліч даних), на відміну від SPMD-моделі, де на кожному процесорі виконуються тільки однакові завдання [12].

Parallel Virtual Machine.

Parallel Virtual Machine (PVM) (дослівно віртуальна паралельна машина) − загальнодоступний програмний пакет, що дозволяє об'єднувати різнорідний набір комп'ютерів в загальний обчислювальний ресурс

(«віртуальну паралельну машину») і надає можливості управління процесами за допомогою механізму передачі повідомлень. Існують реалізації PVM для різноманітних платформ від лептопів до суперкомп'ютерів Cray.

PVM дозволяє користувачам використовувати існуючі апаратні засоби для вирішення проблеми за мінімальних витрат. Сотні сайтів по всьому світу використовують PVM для вирішення важливих наукових, промислових і медичних проблем, PVM також використовується як освітній інструмент для навчання паралельного програмування.

З десятками тисяч користувачів, PVM став де-факто стандартом для розподілених обчислень в усьому світі.

Організація розподілених обчислень.

Методи організації розподілених обчислень дозволяють скористатися перевагами ресурсів, розміщених у корпоративних, глобальних і локальних мережах. Розподілені обчислення зазвичай містять мережеві обчислення в певній формі. Це означає, що програмі, яка виконується на одній комп'ютерній системі (КС) в одній мережі, потрібен деякий апаратний або програмний ресурс, який належить іншій КС в тій самій або віддаленій мережі. Тобто*, методи розподіленого програмування надають доступ до ресурсів, які географічно можуть перебувати на великій відстані один від одного.* Методи розподіленого програмування передбачають спільний доступ до дороговартісних програмних і апаратних ресурсів. Розподілені обчислення можна використовувати для створення певного рівня резервування обчислювальних засобів на випадок аварії.

Найпростішою та найбільш поширеною моделлю розподіленого оброблення даних є *модель типу «клієнт/сервер»*. Зазвичай між клієнтом і сервером існує відношення типу «багато-до-одного», тобто, як правило, один сервер відповідає на запити багатьох клієнтів. Незважаючи на те, що модель типу «клієнт/сервер» – найпоширеніша модель розподіленого програмування, все-таки вона не єдина. Використовуються також *агенти* – раціональні компоненти програмного забезпечення, які характеризуються самонаведенням та автономністю, і можуть постійно перебувати в стані виконання. Агенти співпрацюють в межах груп для колективного виконання певних завдань. У такої моделі не існує конкретного клієнта або сервера. Це модель мережі з рівноправними вузлами (*peer-to-peer*), в якій всі компоненти мають однакові права.

Найбільш поширеними середовищами для паралельного та розподіленого програмування є кластери, SMP- та MPP-комп'ютери.

Кластери – це колекції, що складаються з декількох КС, об'єднаних мережею для створення єдиної логічної системи. З погляду програми така група КС виглядає як одна віртуальна ОС [12].

Під MPP-конфігурацією (процесори з масовим паралелізмом) розуміється один комп'ютерний засіб, що містить сотні процесорів, а під SMP-конфігурацією (симетричний мультипроцесор) – єдина система, в

якій тісно пов'язані процесори спільно використовують загальну пам'ять та інформаційний канал.

На сучасному етапі паралельне багатопроцесорне оброблення є одним з основних напрямків для забезпечення високошвидкісного оброблення інформації. У глобальному аспекті технологія паралельних обчислень розвивається в двох напрямках:

- обчислювальні системи з масовим (переважно, грубозернистим) паралелізмом на базі традиційних процесорів з послідовною системою команд, RISC і сигнальних процесорів тощо;

- нейрокомп’ютинг – нейроподібні обчислювальні системи з масовим дрібнозернистим паралелізмом.

В останні десятиліття було досягнуто значних результатів зі створення обчислювальних систем з паралельною архітектурою першого типу. Разом з тим, відбувався бурхливий розвиток досліджень і розробок зі створення й застосування обчислювальних систем на базі нейроподібних технологій. Причиною підвищеної уваги до нейрокомп’ютингу є властиві йому унікальні потенційні можливості виконання інтелектуальних операцій, а також можливості істотного підвищення критерію продуктивність/вартість порівняно з архітектурами традиційних комп'ютерних засобів. Такі можливості пояснюються наявністю, певною мірою, функціональної та структурної подібності з біологічними прототипами – нейронними мережами головного мозку людини (природній паралелізм).

Основна мета технологій паралелізму – забезпечити умови, що дозволяють комп'ютерними засобами здійснити більший обсяг роботи за той самий період часу [12].

2.6 Технологія GPGPU

Практично будь-яка сучасна платформа виконання програмного коду, повноцінна операційна система або віртуальна машина (наприклад, JAVA машина або .NET framework), що підтримує мультизадачність, містить набір API, призначений для управління потоками і створення паралельних програм. Таким чином, є можливість організувати паралельні обчислення практично будь-якою мовою – від Assembler до скриптових мов типу Perl. Ясно, що проектувати паралельні програми не завжди виправдане рішення з погляду витрат часу і якості коду, оскільки на розробника часто лягає безліч специфічних рутинних завдань зі створення, управління, контролю і забезпечення синхронізації потоків виконання. Йдеться, безумовно, про рішення обчислювально складних задач, оскільки прикладні програми створюються, переважно, використовуючи API платформи. На сьогодні є бібліотеки і мови паралельного програмування, що спрощує безліч проблем, надаючи користувачу механізми для організації паралельних обчис-

лень. Серед них можна відзначити MPI, PVM, мови Cisal, NESL, ZPL, Java, а також розширення різноманітних мов програмування [14].

Останнім часом почали приділяти увагу концепції GPGPU (General-purpose graphics processing units) − технології використання графічного процесора відеокарти для загальних обчислень, які зазвичай виконує центральний процесор.

Необхідність переходу до процесорів загального призначення

Аналіз різних графічних задач показав, що строге закріплення функцій за вершинними і піксельними процесорами неефективне. Дійсно, якщо тривимірні моделі мають насичену геометрію, то, переважно, використовуються вершинні процесори, а якщо модель насичена піксельними ефектами, то основна робота відводиться піксельним процесорам. Це, головним чином, і послужило аргументацією для компанії Nvidia для переходу на процесори загального призначення. Останні здатні виконувати не тільки функції вершинних і піксельних процесорів, але і виконувати різні розрахункові роботи загального призначення.

Аналіз шейдерних програм, що застосовуються Nvidia та ATI, показав неефективність використання обчислювальних ресурсів за векторної архітектури виконавчих блоків. Це призвело до розуміння необхідності переходу в уніфікованих процесорах до скалярних обчислень, доручивши роботу з перетворення векторних операцій на скалярні самому GPU, що і було зроблено компанією Nvidia в графічному процесорі GeForce 8800.

Згідно з архітектурою GeForce 8800 вхідні дані (input stream) процесором обробляються, а його вихід (output stream) йде на вхід іншого процесора для подальшої обробки. Циклічна потокова обробка дозволяє, якщо необхідно, провести повторну обробку даних, що зустрічається досить часто в графічних побудовах, без повторного введення вихідних даних або для їх подальших перетворень.

Для 3D відеоприскорювачів в кінці першої декади XXI століття з’явилися перші технології нерафічних розрахунків. Сучасні відеочипи містять сотні математичних виконавчих блоків, і ця тотужність може використовуватися для значного прискорення безлічі обчислювально інтенсивних додатків. Нинішні покоління GPU мають досить гнучку архітектуру, що разом з високорівневими мовами програмування робить їх значно доступнішими для складних обчислень.

Використання суперкомп'ютерів для паралельних обчислень стає все більш дорогим задоволенням. Крім того, суперкомп'ютери потребують постійної модернізації для підтримки ефективності обчислень. З огляду на це відбувається поступове переведення наукових обчислень на інші платформи. Застосування GPU процесорів дозволяє підвищити ефективність обчислень, використовуючи спеціалізовані бібліотеки. Відеокарти досить дешеві, легко замінні і до певних меж їх кількість можна нарощувати без особливих труднощів.

Останнім часом популярними у використанні були чотири платформи, які реально втілили концепцію GPGPU [14].

AMD FireStream − технологія GPGPU, яка дозволяє програмістам реалізовувати алгоритми, що виконуються на графічних процесорах прискорювачів ATI.

CUDA − технологія GPGPU, яка дозволяє реалізовувати мовою програмування C алгоритми, що виконуються на графічних процесорах прискорювачів GeForce восьмого покоління і старше (GeForce 8 Series, GeForce 9 Series, GeForce 200, 300, 400 Series), Nvidia Quarod і Nvidia Tesla компанії Nvidia. Технологія CUDA розроблена компанією Nvidia. CUDA ToolKit 3.0 (Nvidia) підтримує OpenCL.

DirectCompute − набір інтерфейсів програмування додатків (API) компанії Microsoft є частиною останніх версій DirectX. Він призначений для виконання обчислень загального призначення на графічних процесорах. Безумовно, він може використовуватися і в ігровій практиці. Підтримується компаніями AMD і Nvidia.

DirectCompute, з'явившись в складі DirectX 11, фактично став першою технологією в складі DirectX, яка надала доступ до обчислень загального призначення на графічних процесорах.

Незважаючи на орієнтацію на неграфічні обчислення загального призначення, DirectCompute може використовуватися і в ігровій графіці, наприклад, за рендерінгу тіней, рендерінгу напівпрозорих поверхонь без попереднього сортування, а також під час обробки і фільтрації цифрових зображень, прорахунку алгоритмів ігрового штучного інтелекту і для інших завдань.

OpenCL є мовою програмування задач, пов'язаних з паралельними обчисленнями на різних графічних і центральних процесорах. Мова програмування базується на стандарті C99. OpenCL містить також інтерфейс програмування додатків і забезпечує паралелізм на рівні інструкцій і на рівні даних.

Потрібно зазначити, що безліч додатків з молекулярного моделювання добре пристосовані для розрахунків на відеочипах. На сьогоднішній день GPU пропонують високу продуктивність. Так, наприклад, GPU (Evergreen) позиціонують продуктивність до 10^{12} оп/с. Тому розумне використання тандему (CPU, GPU) дозволить істотно прискорити обробку великих масивів даних.

Основні відмінності між GPU і CPU полягають в архітектурних рішеннях (потокова і топографічна) та організації пам'яті (ієрархічна і модель з максимальною пропускною здатністю).

Однак, моделі програмування GPU (CUDA і OpenCL), які нині широко використовуються є низькорівневими, що потребує від програміста значних зусиль з проектування практично повного сценарію обробки даних, зокрема в трансформації звичайних програм і управління ресурсами.

Типова програма обчислень на CUDA виконує такі дії [14]:

- копіює необхідні дані з оперативної пам'яті CPU в оперативну пам'ять GPU;
- задає розмірність блоків обчислень, їх кількість та ініціалізує процес обчислень на CUDA;
- кожен потік, який з'явився, копіює частину необхідних для виконання блоку даних в швидку пам'ять, що розділяється;
- виконує обчислення;
- копіює результати виконання в основну пам'ять GPU;
- копіює результати з пам'яті GPU в основну оперативну пам'ять комп'ютера.

Технологія загальних обчислень

В обчислювальній моделі CUDA графічний процесор GPU можна розглядати як співпроцесор до CPU з власною пам'яттю, на який передаються обчислення для паралельної обробки великої кількості потоків. Водночас накладні витрати на управління паралельною обробкою на GPU мінімальні порівняно з аналогічною акцією на CPU. Ясно, що чим більше потоків буде оброблятися на графічних процесорах, тим ефективніше вони будуть використовуватися.

Обробка на GPU + CPU здійснюється за схемою, поданою на рис. 2.5. В цьому випадку CUDA надає розробнику ефективних програмних проектів ряд функцій, які можуть виконуватися тільки на CPU, так званий CUDA host API.

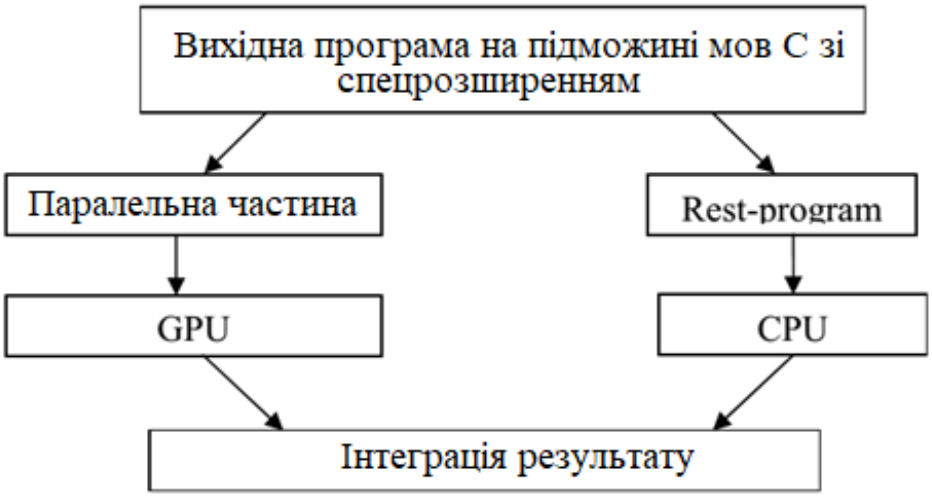


Рисунок 2.5 − Структура обробки програм на тандемі GPU + CPU

Система CUDA має можливість автоматичного розбиття оброблюваної частини на потоки та управління ними. В цьому разі всі потоки організовуються в ієрархію: множина потоків (grid), блоки і окремі потоки.

Ясно, що взаємодію між потоками краще не допускати, щоб вартість обробки була мінімальною. Але якщо така взаємодія необхідна, то в CUDA для цього є два механізми: колективна (shared) пам'ять; бар'єрна синхронізація.

Оскільки обробка потоків йде за технологією SIMD, то фактично виконується одна інструкція, але над різними даними. В цьому випадку вихідна програма розробляється на «урізаному» C (немає операцій введення/виведення, ряд функцій не підтримуються тощо), а відповідні файли мають розширення *.cu*. Якщо деякі функції можуть виконуватися як на CPU, так і на GPU, то відповідні специфікатори host і device можуть використовуватися разом.

Компілятор автоматично згенерує код для обох платформ. Для кожного потоку будуть відомі: індекс потоку всередині блока (threadldx, за замовчуванням вони передбачаються тривимірними) та індекс блока всередині мереж (blockldx). Блок потоків виконується на мультипроцесорі пулами, кожен з яких містить, як правило, 32 потоки. У цьому випадку всередині пулу одночасно може виконуватися тільки одна інструкція. Пул розбивається на частини, кратні кількості процесорів в мультипроцесорі і виконуються послідовно. Обмін даними між завданнями всередині блока здійснюється через загальну пам'ять, що розділяється.

Безумовно, можливі різні евристики зі взаємодії процесів, але вони враховують особливості реальної апаратури і власні системні напрацювання. Як останні можна використовувати, наприклад, середовище реалізації Common Language Runtime (CLR) платформи .NET. Це дозволяє зробити реалізацію перетворення алгоритмів розв'язання задач на GPU, незалежну від платформи і мови програмування. Щоб виконати паралельні фрагменти на GPU необхідно [14]:

- завантажити модуль з необхідними функціями для відеокарти;
- виділити необхідний обсяг пам'яті на GPU;
- скопіювати дані з оперативної пам'яті в пам'ять відеокарти;
- проініціювати функції з завантаженого модуля, вказавши ступінь розпаралелювання;
- обробити дані;
- скопіювати результати з пам'яті GPU в пам'ять CPU.

Вимоги до задач для ефективного виконання на GPU

Аналіз технології обробки задач на відеокартах, архітектурних рішень CPU і GPU, а також проведення ряду експериментів на такому тандемі дозволяє стверджувати, що насамперед потрібно пропонувати задачі, які можна розпаралелити на сотні потоків. Особливо вагоме прискорення можна отримати, якщо одні й ті самі інструкції застосовуються до величезних масивів даних.

Наступною вимогою можна назвати відсутність взаємодій між потоками, які обробляються, або «слабка» взаємодія. Серед інших властивостей задач (програм) для обробки на GPU можна назвати:

- мінімальна кількість складних для обробки операцій: ділення, піднесення до від'ємного степеня тощо;
- відсутність в алгоритмах множинного розгалуження;
- невеликий обсяг даних, переданих до відеокарти і від неї в оперативну пам'ять CPU.

Безумовно, іноді зазначені вимоги можна частково обійти за рахунок ретельного аналізу алгоритму розв'язання задачі і створення додаткових засобів, що знижують вплив зазначених та інших вимог на час повного циклу вирішення задач. Але все це треба робити надзвичайно коректно [14].

Питання для самоконтролю

1. Охарактеризуйте організацію паралельних обчислень з використанням сучасних технологій.
2. Наведіть класифікацію обчислювальних систем за Флінном.
3. Наведіть розширену класифікацію обчислювальних систем М. Флінна із врахуванням паралельних обчислювальних систем.
4. Охарактеризуйте сучасну класифікацію паралельних обчислювальних систем.
5. Охарактеризуйте та наведіть рівні паралелізму для паралельної обробки інформації.
6. Яке практичне значення комбінування паралельних та розподілених обчислень?
7. Розкрийте суть складових «декомпозиція», «зв'язок» та «синхронізація», які є основою процесу проектування паралельних і розподілених комп'ютерних систем та їх програмно-апаратного забезпечення.
8. Охарактеризуйте високопродуктивні обчислювальні системи з гібридною архітектурою.
9. Охарактеризуйте технологію GPGPU.
10. Наведіть вимоги до задач для ефективного виконання на GPU.

3 НЕЙРОПОДІБНІ МЕРЕЖЕВІ ТЕХНОЛОГІЇ

3.1 Методика організації обчислювальних процесів і структурно-функціональне забезпечення штучних нейронних мереж

Нейрон являє собою одиницю обробки інформації в нейронній мережі. На блок-схемі (рис. 3.1) показано модель нейрона, що лежить в основі штучних нейронних мереж. У цій моделі можна виділити три основні елементи [15].

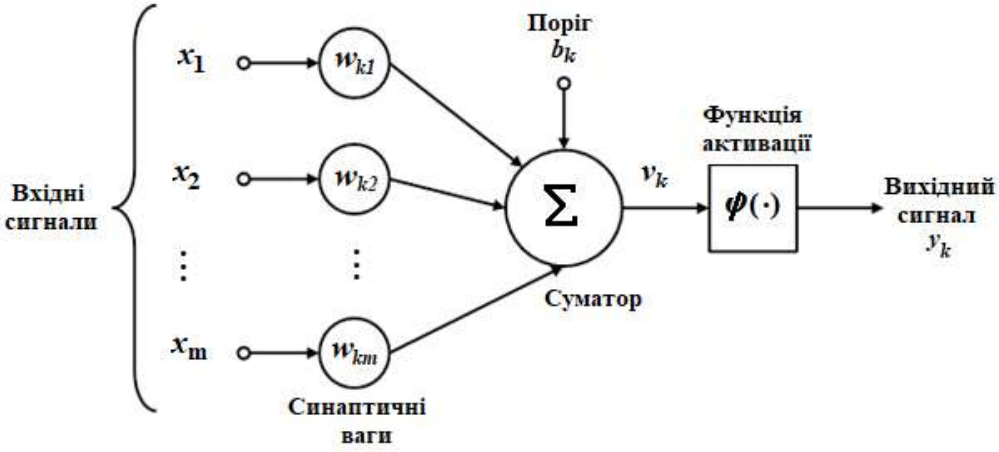


Рисунок 3.1 − Нелінійна модель нейрона

1. Набір синапсів або зв'язків, кожен з яких характеризується своєю вагою або силою. Зокрема сигнал x_j на вході синапсу j, пов'язаного з нейроном k, множиться на вагу w_{kj}. Важливо звернути увагу на те, в якому порядку вказано індекси синаптичної ваги w_{kj}. Перший індекс відноситься до розглянутого нейрона, а другий − до вхідного закінчення синапсу, з яким пов'язана ця вага. На відміну від синапсів мозку синаптична вага штучного нейрона може мати як позитивні, так і негативні закінчення.

2. Суматор додає вхідні сигнали, зважені щодо відповідних синапсів нейрона. Цю операцію можна описати як лінійну комбінацію.

3. Функція активації обмежує амплітуду вихідного сигналу нейрона. Ця функція також називається функцією стиснення. Зазвичай нормалізований діапазон амплітуд виходу нейрона лежить в інтервалі [0,1] або [-1,1].

У моделі нейрона, показаній на рис. 3.1, введено пороговий елемент, який позначено символом b_k. Ця величина відображає збільшення або зменшення вхідного сигналу, що подається на функцію активації.

В математичному поданні функціонування нейрона k можна описати такою парою рівнянь [15]:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \,, \tag{3.1}$$

$$y_k = \varphi(u_k + b_k)\,, \tag{3.2}$$

де $x_1, x_2, ..., x_m$ – вхідні сигнали;

$w_{k1}, w_{k2}, ..., w_{km}$ – синаптичні ваги нейрона k;

u_k – лінійна комбінація вхідних впливів;

b_k – поріг,

$\varphi(\cdot)$ – функция активации;

y_k – вихідний сигнал нейрона.

Використання порога b_k забезпечує ефект афінного перетворення виходу лінійного сумматора u_k. У моделі, показаної на рис. 3.1, постсинаптичний потенціал обчислюється таким чином [15]:

$$\upsilon_k = u_k + b_k\,. \tag{3.3}$$

Зокрема, залежно від того, якого значення набуває поріг b_k, додатного або від'ємного, індуковане локальне поле або потенціал активації υ_k нейрона k змінюється так, як показано на рис. 3.2. Надалі будемо використовувати термін «індуковане локальне поле». Зверніть увагу на результат афінного перетворення. Графік υ_k вже не проходить через початок координат, як графік u_k.

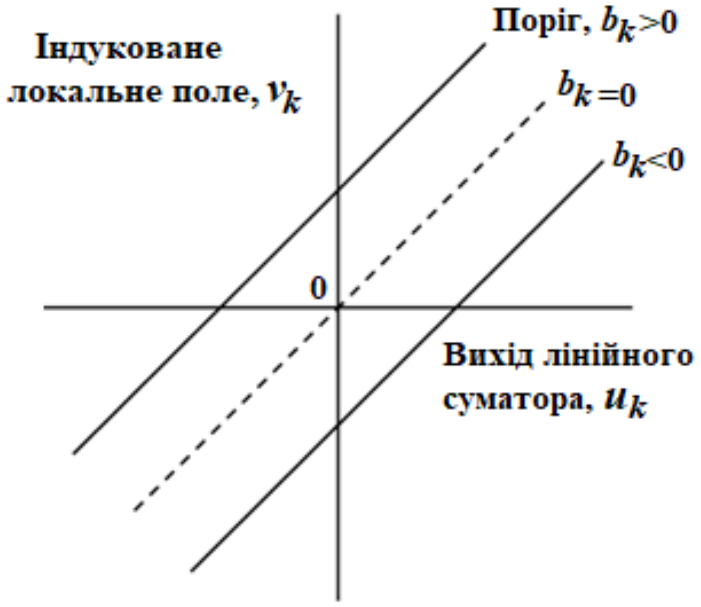


Рисунок 3.2 – Афінне перетворення, викликане наявністю порога, в цьому випадку в точці $u_k = 0$, $\upsilon_k = b_k$

Поріг b_k є зовнішнім параметром штучного нейрона k. Його присутність ми бачимо у виразі (3.2). Беручи до уваги вираз (3.3), формули (3.1), (3.2) можна перетворити до такого вигляду [15]:

$$\upsilon_k = \sum_{j=0}^{m} w_{kj} x_j \,, \tag{3.4}$$

$$y_k = \varphi(\upsilon_k) \,. \tag{3.5}$$

У виразі (3.4) додався новий синапс. Його вхідний сигнал дорівнює:

$$x_0 = +1, \tag{3.6}$$

А його вага:

$$w_{k0} = b_k \,. \tag{3.7}$$

Це дозволило трансформувати модель нейрона до вигляду, показаного на рис. 3.3. На цьому рисунку видно, що внаслідок введення порогу додається новий вхідний сигнал фіксованої величини +1, а також з'являється нова синаптична вага, яка дорівнює пороговому значенню b_k. Хоча моделі, показані на рис. 3.1 і рис. 3.3, зовні абсолютно несхожі, математично вони еквівалентні.

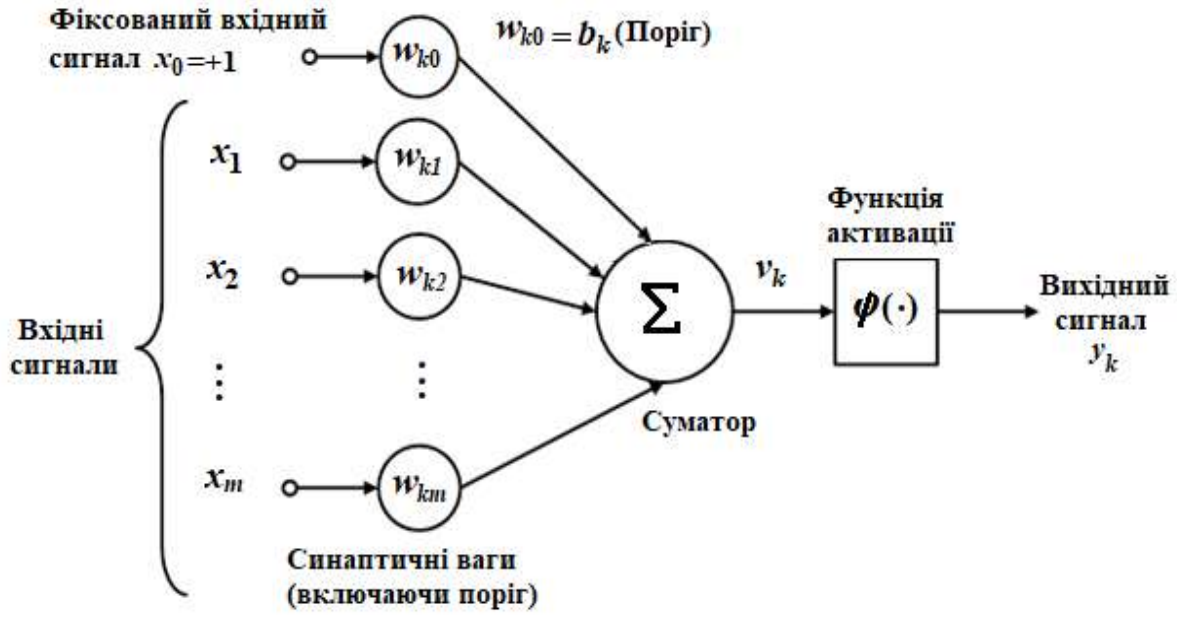


Рисунок 3.3 − Інша нелінійна модель нейрона

Структура нейронних мереж тісно пов'язана з використовуваними алгоритмами навчання. У загальному випадку можна виділити три фундаментальні класи нейромережевих архітектур.

Одношарові мережі прямого поширення

У багатошаровій нейронній мережі нейрони розташовуються шарами. У найпростішому випадку в такій мережі існує вхідний шар вузлів джерела, інформація від якого передається на вихідний шар нейронів, але не навпаки. Така мережа називається мережею прямого поширення або ациклічною мережею. На рис. 3.4 показано структуру такої мережі для випадку чотирьох вузлів в кожному з шарів. Така нейронна мережа називається одношаровою, в цьому випадку під єдиним шаром мають на увазі шар обчислювальних елементів (нейронів). Під час підрахунку числа шарів не беруть до уваги вузли джерела, оскільки вони не виконують ніяких обчислень [15].

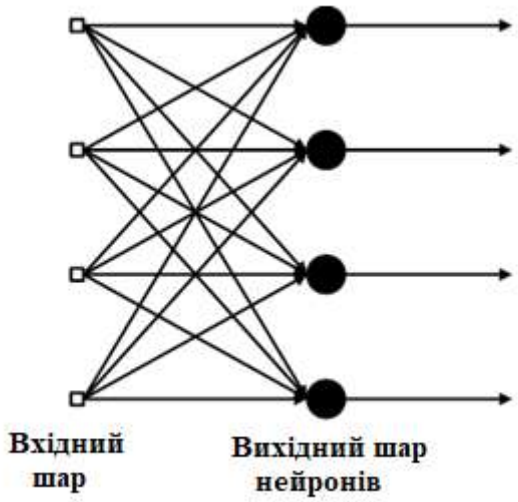


Рисунок 3.4 − Мережа прямого поширення з одним шаром нейронів

Багатошарові мережі прямого поширення

Інший клас нейронних мереж прямого поширення характеризується наявністю одного або декількох прихованих шарів, вузли яких називаються прихованими нейронами або прихованими елементами. Функція останніх полягає в посередництві між зовнішнім вхідним сигналом і виходом нейронної мережі. Додаючи один або кілька прихованих шарів, ми можемо виділити статистики високого порядку. Така мережа дозволяє виділяти глобальні властивості даних за допомогою локальних з'єднань за рахунок наявності додаткових синаптичних зв'язків і підвищення рівня взаємодії нейронів. Здатність прихованих нейронів виділяти статистичні залежності високого порядку особливо істотна, коли розмір вхідного шару досить великий.

Вузли джерела вхідного шару мережі формують відповідні елементи шаблона активації (вхідний вектор), з яких складається вхідний сигнал, що надходить на нейрони (обчислювальні елементи) другого шару (тобто першого прихованого шару). Вихідні сигнали другого шару використовуються як вхідні для третього шару і т. д. Зазвичай нейрони кожного з шарів мережі використовують як вхідні сигнали вихідні сигнали нейронів тільки попереднього шару. Набір вихідних сигналів нейронів вихідного (останнього) шару мережі визначає загальний відгук мережі на цей вхідний образ, сформований вузлами джерела вхідного (першого) шару. Мережа, показана на рис. 3.5, називається мережею 10-4-2, оскільки вона має 10 вхідних, 4 прихованих та 2 вихідних нейрони. У загальному випадку мережа прямого поширення з m входами, h_1 нейронами першого прихованого шару, h_2 нейронами другого прихованого шару і q нейронами вихідного шару називається мережею $m-h_1-h_2-q$ [15].

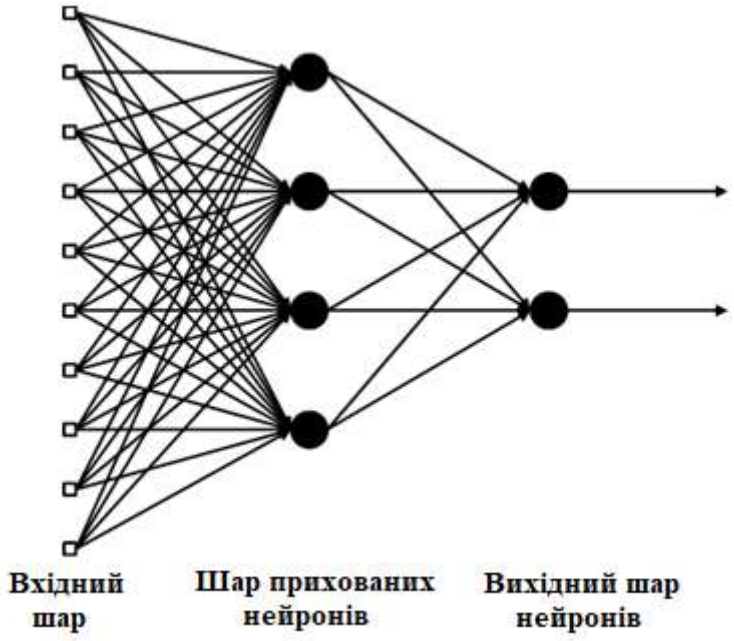


Рисунок 3.5 − Повнозв'язна мережа прямого поширення з одним прихованим і одним вихідним шаром

Нейронна мережа, показана на рис. 3.5, вважається повнозв'язною в тому розумінні, що всі вузли кожного конкретного шару з'єднані з усіма вузлами суміжних шарів. Якщо деякі з синаптичних зв'язків відсутні, така мережа називається неповнозв'язною.

Рекурентні мережі

Рекурентна нейронна мережа відрізняється від мережі прямого поширення наявністю принаймні одного зворотного зв'язку. Наприклад, рекуре-

нтная мережа може складатися з єдиного шару нейронів, кожен з яких спрямовує свій вихідний сигнал на входи всіх інших нейронів шару. Архітектуру такої нейронної мережі показано на рис. 3.6 [15].

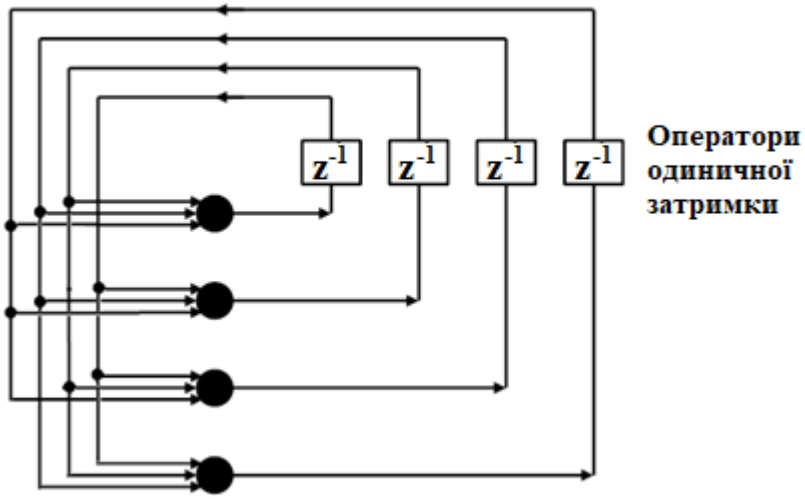


Рисунок 3.6 − Рекурентна мережа без прихованих нейронів і зворотних зв'язків нейронів з самими собою

Крім того, у наведеній структурі відсутні зворотні зв'язки нейронів з самими собою. Рекурентна мережа, показана на рис. 3.6, не має прихованих нейронів. На рис. 3.7 показано інший клас рекурентних мереж − з прихованими нейронами. Тут зворотні зв'язки виходять як з прихованих, так і з вихідних нейронів.

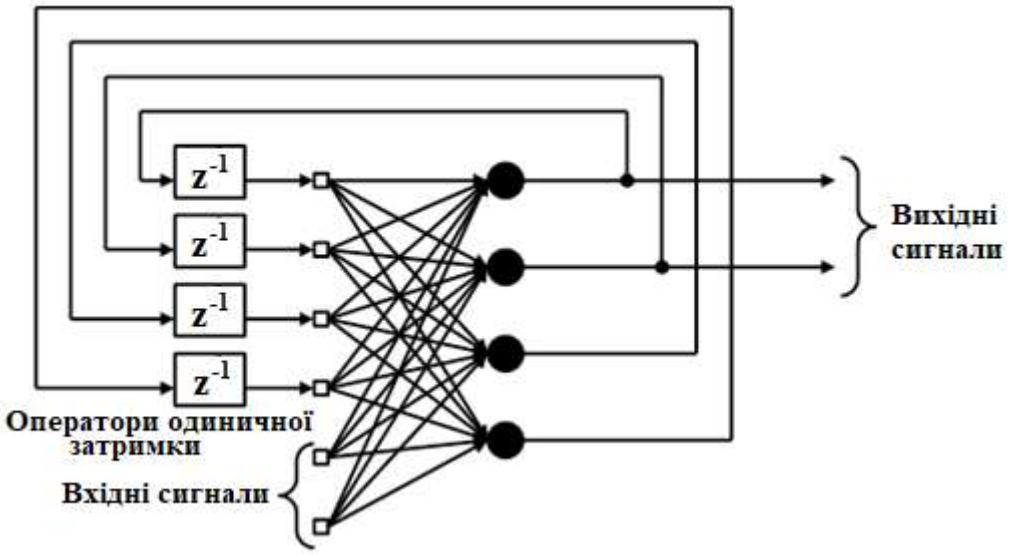


Рисунок 3.7 − Рекурентна мережа з прихованими нейронами

Наявність зворотного зв'язку в мережах, показаних на рис. 3.6 і рис. 3.7, безпосередньо впливає на здатність таких мереж до навчання і на їх продуктивність. Більш того, під зворотним зв'язком мають на увазі використання елементів одиничної затримки (вони позначені як z^{-1}), що призводить до нелінійної динамічної поведінки, якщо, звичайно, в мережі є нелінійні нейрони [15].

3.2 Сучасні інформаційні технології моделювання штучних нейронних мереж

Нині відомо більше 200 нейропакетів, що випускаються рядом фірм та окремими дослідниками і дозволяють конструювати, навчати і використовувати нейронні мережі для вирішення практичних завдань [16].

Розглянемо деякі з них [16]:

- пакет *Neural 10,* розроблений в 1992 році, використовує одну нейромережевому парадигму – двошарову мережу прямого поширення з алгоритмом навчання (зворотного поширення помилки). Активаційна функція нейронів прихованого шару − сигмоїда, а вихідних нейронів − лінійна.

- програма *Neuro Pro* розповсюджується вільно альфа-версією нейромережевого програмного продукту для роботи зі штучними нейронними мережами та вилучення знань з таблиць даних за допомогою нейронних мереж в середовищі Windows.

Можливості програми такі:

- робота з файлами в форматах * .dbf і * .db;
- створення нейронних мереж з кількістю шарів до 10 і нейронів в шарі до 100;
- використання нелінійної сигмоїдної функції f (A) = A / (| A | + c);
- вирішення задач прогнозування (передбачення значень кількісних вихідних ознак) і класифікації (прогноз станів якісних вихідних ознак). Нейромережа може мати кілька вихідних сигналів (вирішувати одночасно кілька завдань прогнозування і класифікації) для кожного з вихідних сигналів можуть бути встановлені свої вимоги до точності виконання завдання;
- навчання нейронної мережі із застосуванням одного з таких методів градієнтної оптимізації (градієнт обчислюється за принципом подвійного функціонування): Метод найшвидшого спуску. Модифікований ParTan-метод. Метод сполучених градієнтів. Квазіньютоновский BFGS-метод;
- тестування нейронної мережі, отримання статистичної інформації про точність розв'язання задачі. Обчислення і відображення значимості вхідних сигналів мережі, збереження значень показників значущості в файлі на диску. Внесення випадкових збурень в ваги синапсів мережі. Спрощення (контрастування) нейронної мережі: Скорочення числа вхідних

сигналів мережі. Скорочення числа нейронів мережі. Рівномірне проріджування структури синапсів мережі. Скорочення числа синапсів мережі. Скорочення числа неоднорідних входів (порогів) нейронів мережі. Бінаризація ваг синапсів мережі (приведення ваг синапсів і порогових входів до кінцевого набору виділених значень). Можливий вибір з 4-х наборів виділених значень;

- генерація вербального опису нейронної мережі. Вербальний опис може редагуватися і зберігатися в файлі на диску. Генерація вербального Опису нейронної мережі.

• Нейропакет QwikNet32 призначений для роботи в середовищі Windows і реалізує один тип нейронної мережі − багатошарову мережу прямого поширення з числом прихованих шарів до п'яти і з набором з шести алгоритмів навчання.

Можливості нейропакета QwikNet32 [16]:

- застосування чотирьох видів функцій активації (сигмоїдної, гіперболічний тангенс, лінійна, функція Гаусса);
- використання одного з шести алгоритмів навчання;
- навчання з перехресним перетином (навчальна вибірка ділиться автоматично на два набори: 90% − для навчання і 10% − для тестування і перевірки якості навчання);
- рандомізація значень ваг синапсів перед навчанням;
- установлення коефіцієнта швидкості навчання;
- графічне подання результатів навчання;
- виведення повідомлення про коректність навчання для всіх виходів мережі.

• Нейропакет Neural Planner має програмну оболонку, що дозволяє моделювати нейронні мережі різної конфігурації.

Можливості нейропакета Neural Planner такі [16]:

- додавання і видалення вхідного і вихідного нейронів, синапсу, з'єднання і роз'єднання двох шарів нейронів;
- установлення параметрів активаційної функції і кількості циклів навчання;
- вибір виду графіка зміни середньої помилки під час навчання мережі;
- відображення інформації про будь-який обраний в мережі нейрон;
- відображення середньої та цільової помилки;
- редагування файлів навчальної вибірки за допомогою електронних таблиць MS Excel.

• *Пакет програм NeuralWorks Professional II Plus* є одним з останніх версій програмного продукту NeuralWorks, розробленого фірмою

NeuralWare. Пакет містить програмні моделі десятків архітектур нейронних мереж.

- *Пакет програм ExploreNet 3000 є* розробкою фірми HNC, заснованої професором Робертом Хехт-Нільсеном. Пакет надає широкі можливості з моделювання і управління даними. Як прискорювач використовуються апаратні розробки фірми HNC – нейропроцесори ANZA і ANZA +, які є одними з перших апаратних рішень. Фірма запропонувала також засіб для розробки прикладних програм – спеціалізовану мову програмування AXON, основану на базі мови С.

- *Оболонка NeuroShell*. Перевагою цієї програми є сумісність з популярним пакетом управління даними MicroSoft Excel, що робить продукт зручним для масового використання.

- Пакет *Neuro Office* призначений для проектування інтелектуальних програмних модулів, побудованих на основі нейронних мереж з ядерною організацією. Результатом проектування є навчена нейронна мережа з програмним інтерфейсом, відповідним моделі багатокомпонентних об'єктів.

- *Нейропакет NeuralWorks Professional* є потужним засобом для моделювання нейронних мереж. У ньому реалізовано 28 нейронних парадигм, а також велика кількість алгоритмів навчання. Є хороша система візуалізації даних: структури нейронної мережі, зміни помилки навчання, зміни ваг і їх кореляції в процесі навчання.

- *Пакет NeuroShell 2* (Фірма Neuron Data) використовує правила для попередньої обробки інформації, яка потім передається в нейронну мережу. Отримана на виході нейронної мережі інформація також може бути оброблена за допомогою системи правил.

- *Нейропакет BrainMaker Pro* є простим нейропакетом для моделювання багатошарових нейронних мереж, яких навчають за допомогою алгоритму зворотного поширення помилки. Основною його перевагою є велике число параметрів налаштування алгоритму навчання [16].

- STATISTICA Automated Neural Networks (SANN) є одним з найбільш передових і найефективніших нейромережевих продуктів на ринку. Він пропонує безліч унікальних переваг і багатих можливостей. Наприклад, унікальні можливості інструменту автоматичного нейромережевого пошуку, Автоматизована нейронна мережа (АНМ), дозволяють використовувати систему не тільки експертам з нейронних мереж, але і новачкам в області нейромережевих обчислень [17].

Можливості такі:

- пре- і пост-процесування, включно вибір даних, кодування номінальних значень, шкалювання, нормалізацію, видалення пропущених даних з інтерпретацією для класифікації, регресію і задання часових рядів;
- виняткова простота у використанні, крім того, неперевершена аналітична потужність;
- найсучасніші, оптимізовані і потужні алгоритми навчання мережі (включно методи спряжених градієнтів, алгоритм Левенберга-Марквардта, BFGS, алгоритм Кохонена); повний контроль над усіма параметрами, що впливають на якість мережі, такими як функції активації і помилок, складність мережі;
- підтримка ансамблів нейромереж і нейромережевих архітектур практично необмеженого розміру;
- багаті графічні і статистичні можливості, які полегшують інтерактивний дослідний аналіз;
- повна інтеграція з системою STATISTICA; всі результати, графіки, звіти і т. д. можуть бути в подальшому модифіковані за допомогою потужних графічних і аналітичних інструментів STATISTICA (наприклад, для проведення аналізу передбачених залишків, створення докладного звіту і под.);
- повна інтеграція з потужними автоматичними інструментами STATISTICA; запис повноцінних макросів для будь-яких аналізів; створення власних нейромережевих аналізів і додатків за допомогою STATISTICA Visual Basic, виклик STATISTICA Automated Neural Networks з усіх програм, що підтримує технологію COM (наприклад, автоматичне проведення нейромережевого аналізу в таблиці MS Excel або об'єднання декількох призначених для користувача додатків, написаних мовами C, C++, C #, Java і т. д.);
- вибір найбільш популярних мережевих архітектур;
- наявний інструмент Автоматичного Мережевого Пошуку, що дозволяє в автоматичному режимі будувати різні нейромережеві архітектури і регулювати їх складність;
- збереження найкращих нейронних мереж.

• Пакет MATLAB полегшує процедуру конструювання, навчання та використання ШНМ. Користувачу не потрібно досконало розуміти механізм створення нейромережі, варто лише керуватися запитами програми. Проте не можна починати роботу, не будучи обізнаним з функціонуванням нейромережі [18].

Перед початком роботи в середовищі будь-якої програми потрібно:

– визначити загальний тип виконуваного завдання і відповідно до нього обрати структуру або архітектуру нейронної мережі; водночас потрібно зупинитися на найефективнішій і перевірити очікуваний результат;

– підготувати набір навчальних і тестових даних; перевірити, щоб приклади навчальної множини не повторювалися в тестовій, оскільки мережа просто запам'ятає результат і відтворить його;

– вибрати кількість вхідних та вихідних нейронів, прихованих шарів; для цього потрібно чітко визначити кількість вхідних параметрів – це і буде кількість вхідних нейронів, а вихідні параметри відповідно визначать кількість вихідних нейронів. Кількість прихованих шарів нейронної мережі можна скоригувати і в процесі роботи саме тут потрібно провести ряд експериментів, щоб визначити найоптимальніший варіант.

Програмний пакет MATLAB від компанії розробника MathWorks – один з найпотужніших інструментів оброблення даних на ринку програмних продуктів. Штучні нейронні мережі в середовищі MATLAB можна проектувати як за допомогою спеціального вбудованого пакета NNTool, так і безпосередньо в командному вікні. Попри всі переваги програми, варто відзначити й недоліки – *велику ціну програмного пакета та складність процесу функціонування.*

MATLAB – відомий інструмент аналізу багатовимірних даних, одним з вбудованих програмних пакетів якого є Network Data Manager. Програма дозволяє створювати нейромережі різних типів, використовувати декілька функцій активацій та алгоритмів навчання, обирати кількість прихованих шарів та кількість нейронів у них. Після створення мережі в програмі NNTool можна протестувати, переглянути графік, що ілюструє навчальний процес, та оцінити помилку, обчислену програмою.

Програма MATLAB корисна своїми можливостями створення індивідуальних, нестандартних мереж, проте досить складна для використання, тому для отримання задовільних результатів, потрібно мати досвід створення нейронних мереж.

Програма MATLAB дає змогу переглядати модель створеної штучної нейронної мережі, така модель досить спрощена, проте вона дозволяє візуально оцінювати кількість нейронів, функції активації та розміщення вагових коефіцієнтів.

За допомогою програми можна зображати процес навчання у вигляді графіка, а отже, користувач може визначити, як проходив процес навчання та скільки циклів навчання знадобилося, щоб помилка навчальної множини досягла потрібного рівня.

Важливою перевагою використання програми MATLAB є можливість переглядати та коригувати ваги штучної нейронної мережі. Більшість програм нейромережевого моделювання позбавлені цієї важливої функції.

Програмний пакет MATLAB як один із найпотужніших інструментів оброблення даних надає основні можливості для створення моделей ШНМ. Основними *перевагами* використання програми є можливість створення індивідуальних архітектур та зміни ваг нейронів. *Недоліки* використання цього програмного пакета для побудови ШНМ: висока собівартість про-

грами, складність процесу функціонування, обмежена кількість регульованих параметрів для певних структур нейромереж

Перевагою вищеперерахованих систем є простота створення і зрозумілість процесу виведення, відсутність проблем із внесенням змін. Як недолік можна відзначити високу вартість програмного продукту. Нейропакети, перераховані вище, є відносно дорогими і призначені для професійного використання.

3.3 Організація обчислювальних процесів у нейроподібних паралельно-ієрархічних обчислювальних системах

Паралельно-ієрархічне перетворення та особливості організації паралельно-ієрархічного обчислювального процесу

Аналіз принципів природного та штучного паралелізму

Природний паралелізм безпосередньо пов'язаний із нейроподібною схемою обробки інформації, головною перевагою якої є використання наслідків динаміки багаторівневої паралельної взаємодії інформаційних сигналів на різних рівнях ієрархії нейронної мережі, що дозволяє поєднати такі відомі природні особливості організації обчислень у корі головного мозку (ГМ): топографічний характер відображення, одночасність (паралельність) дії сигналів, мозаїчність структури кори, грубу ієрархічність кори, просторово корельований у часі механізм сприйняття та навчання.

У роботі про нейронні механізми зору людини учений Д. Х'юбел відзначає, що за зорового сприйняття розрізняють два процеси, які відбуваються разом. Перший із них має ієрархічний характер, а другий полягає в дивергенції нейронних шляхів. На скільки поширені процеси дивергенції і конвергенції сигналів можна судити з того, як аксон майже кожної нейронної клітини цього рівня розбивається на гілки під час підходу до наступного рівня, і закінчується в декількох або багатьох клітинах. І навпаки, нейронна клітина довільного рівня, крім першого, має синаптичні входи від декількох або багатьох нейронних клітин попереднього рівня [12].

Принцип паралельно-ієрархічного перетворення

Принцип паралельно-ієрархічного перетворення (ППП) частково враховує підтверджену результатами нейронаук гіпотезу про ієрархічну організацію зв’язків між структурами головного мозку, суть якої ґрунтується на тому, що «передача збудження між структурами, тобто їх активація, може відбуватись не лише за вертикальними (ієрархічним прямим та зворотним) зв’язками, але також за горизонтальними – у межах одного і того самого поля». Відповідно, врахування такої нейроподібної схеми обробки інформації в паралельно-ієрархічному перетворенні теж сприяє ефективній реалізації моделей та програмно-апаратних засобів паралельно-ієрархічних

обчислювальних систем (ПІОС) для високопродуктивної обробки надвеликих масивів інформації.

Паралельно-ієрархічне перетворення – це підхід до створення обчислювального середовища – паралельно-ієрархічної (ПІ) мережі, яка досліджена у вигляді моделі нейроподібної схеми обробки інформації. Підхід має ряд переваг порівняно з іншими методами формування нейроподібного середовища (наприклад, відомими методами формування штучних нейронних мереж). Головною перевагою підходу є використання наслідків динаміки багаторівневої паралельної взаємодії інформаційних сигналів на різних рівнях ієрархії нейроподібної мережі, що дозволяє поєднати такі відомі природні особливості організації обчислень у корі головного мозку (ГМ): топографічний характер відображення, одночасність (паралельність) дії сигналів, мозаїчність структури кори, грубу ієрархічність кори, просторово корельований у часі механізм сприйняття та навчання. Формування багатоетапної ПІ мережі припускає процес послідовного перетворення корельованих і утворення декорельованих у часі елементів нейронної мережі під час переходу її з одного стійкого стану в інший [12].

Суть пірамідального підходу полягає в одночасному використанні під час аналізу послідовності масивів даних на різноманітних рівнях ієрархії. Це дозволяє реалізувати стратегію від «загального до часткового», що дає можливість реалізувати концепцію нейроподібної обробки. Кожний елемент піраміди інформаційного поля характеризується трьома координатами *(i,j,k)*, де *i* – рядок, *j* – стовпець, *k* – рівень.

Таким чином, принцип побудови паралельно-ієрархічної структури даних можна визначити як послідовність операцій над множинами масивів даних, що утворюють множини інформаційних полів різноманітних рівнів ієрархії, взаємодія між якими здійснюється пірамідальною ієрархічною структурою і реалізується на основі мережної архітектури.

Структурну схему взаємодії інформаційних потоків у паралельно-ієрархічній обчислювальній структурі подано на рис. 3.8; в ній обробляється множина вхідних потоків даних на різноманітних (*k*) ієрархічних рівнях. Кожний рівень являє собою сукупність процесорних елементів (ПЕ), які функціонують у чітко фіксовані моменти часу (t_j).

Принцип організації паралельно-ієрархічного обчислювального процесу можна описати, використовуючи моделі PRAM, SPMD (SIMD) і MPMD (MIMD). Оскільки вказані схеми та моделі успішно використовуються для реалізації практичних дрібно- та середньомасштабних додатків (за винятком паралельних обчислень більш високого ступеня організації).

Наукові дослідження концентруються на розробці нейроемулятора – системи, побудованої на базі каскадного з'єднання універсальних SISD-, SIMD- або MISD-процесорів (наприклад, Intel, AMD, Sparc, Alpha, Power PC та ін.), яка реалізує типові нейрооперації (зважене підсумовування й нелінійне перетворення) на програмному рівні. Вирішення поставленої за-

дачі можливе лише за умови коректного та обґрунтованого вибору нейроприскорювача – нейрокомп'ютера, реалізованого у вигляді карти або модуля з розпаралелюванням операцій на апаратному рівні або ж конструктивно-автономної системи. Нейрокомп'ютери, виготовлені у вигляді карт (віртуальні нейрокомп'ютери), як правило, призначені для встановлення у слот розширення комп'ютерної системи (стандартного ПК) [12].

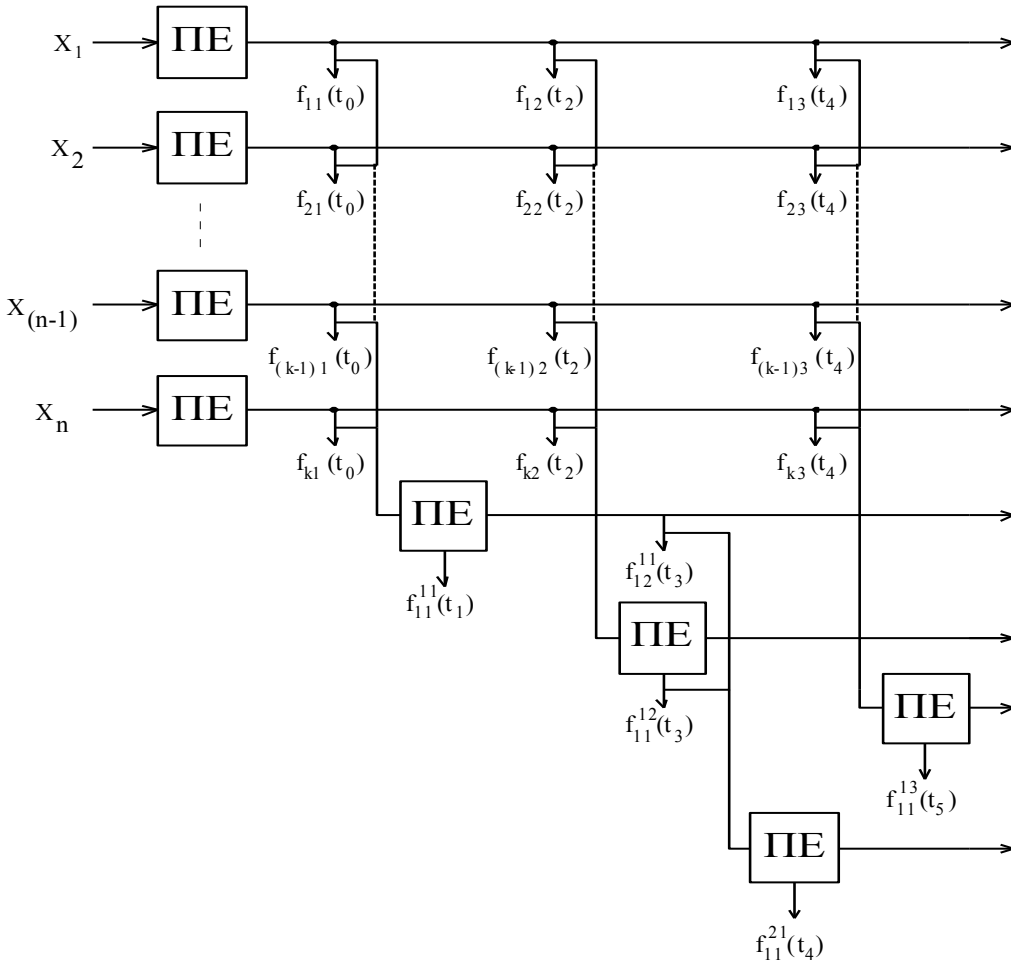


Рисунок 3.8 − Структурна схема взаємодії інформаційних потоків у ПІ мережі

Нині в ролі нейроприскорювача як апаратної платформи для реалізації масштабних нейронних та нейропобідних паралельно-ієрархічних мереж доцільно застосовувати GPGPU – General-Purpose computation on Graphic Processing Units (обчислення загального призначення на відеоадаптерах). GPGPU – це фактично використання потужного відеоадаптера для виконання спеціалізованих, зокрема паралельних, обчислень, які зазвичай виконуються на CPU в ПК. Оскільки сучасні технології побудови відеоадаптерів дозволяють використання 128-, 256-ядерних і більше спецпроцесорів, порівняно із сучасними багатоядерними мультимедійними CPU, то застосування їх для нейроемуляції різних топологій масштабних нейронних та

нейроподібних паралельно-ієрархічних мереж є актуальним та перспективним.

В контексті програмної реалізації здійснюється робота із створення нейропакета для реалізації різних топологій масштабних нейронних та нейроподібних паралельно-ієрархічних мереж і можливості прорахунку їх на GPU (реалізація процесів паралельно-ієрархічної обробки інформації та відповідних методів навчання). Під топологією розуміємо певний набір шарів мережі, відповідно пов'язаних між собою. Кількість шарів мережі, зв'язки між ними, кількість нейронів у шарі, функція активації шару, зв'язки між нейронами різних шарів задаються користувачем. Одна із ключових можливостей програмного продукту – гнучкість у створенні топології; що робить можливим реалізацію масштабних паралельно-ієрархічних та ієрарх-ієрархічних нейропобідних мереж.

Розглянемо методологічні особливості архітектури відеоадаптерів для програмної реалізації паралельних обчислень (обчислень загального призначення) на низькому рівні програмування.

На рис. 3.9 зображено спрощену модель відеоадаптера. Кількість обчислювальних блоків у сучасних відеоадаптерах набагато більше двох. Кількість ядер у кожному обчислювальному блоці також набагато вища [12].

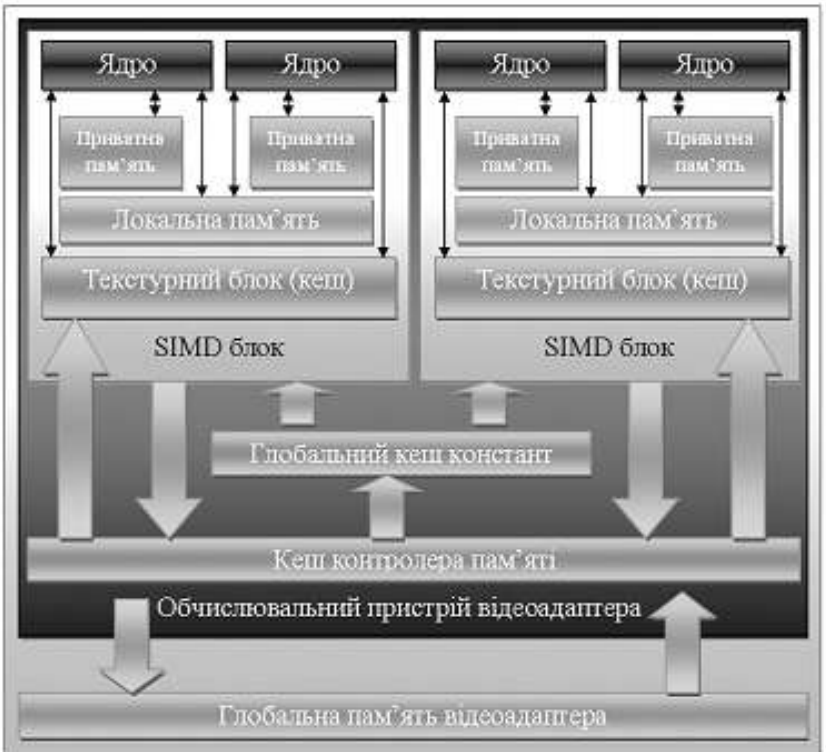


Рисунок 3.9 − Узагальнена модель відеоадаптера

Локальна пам’ять використовується лише під час обчислень загального призначення для синхронізації потоків і реалізована не в усіх відеоадаптерах. Текстурні блоки, яких на кожен SIMD-блок припадає до 4-ох, мають

власний кеш (проміжний буфер із швидким доступом, що містить копію тієї інформації, яка зберігається в пам’яті з менш швидким доступом, але з найбільшою ймовірністю може бути звідти запитаний) та дозволяють завантажувати дані з глобальної пам’яті в приватну, яка належить лише конкретному ядру SIMD-блока («керуючи» ці дані у власному кеші). Усі операції зчитування та записування у глобальну пам’ять виконуються через контролер пам’яті і таким чином зберігаються у кеш контролера пам’яті.

Таким чином, принцип паралелізму реалізовано в сучасних відеоадаптерах на багатьох рівнях. У паралельно працюючих над різними задачами (потоками інструкцій) SIMD-блоках виконується паралельна обробка даних однієї задачі. У сучасних відеоадаптерах зазвичай використовується від 2 до 8 контролерів пам’яті, що дозволяє збільшити пропускну спроможність пам’яті.

Узагальнена модель програмування GPU

Перед виконанням програми її константи завантажуються у «глобальний кеш констант». Після цього спеціальний диспетчер виділяє певній програмі певну кількість SIMD-блоків, на яких вона буде виконуватися. Усі ядра у SIMD-блоці виконують одні і ті самі операції, але над різними операндами (single instruction multiply data), тобто розділяють потік інструкцій. Кожне ядро одночасно може обробляти декілька потоків, кількість яких обмежена лише розміром локальної пам’яті ядра. Під одночасною обробкою розуміється одночасне розміщення даних у пам’яті та почергове виконання інструкцій з цих потоків, що дозволяє маскувати затримки під час зчитування з глобальної пам’яті.

На рис. 3.10 зображено узагальнену модель програмування відеоадаптерів [12].

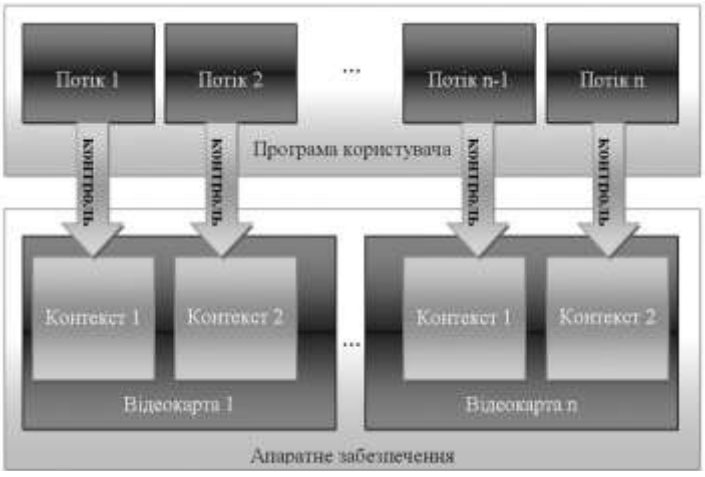


Рисунок 3.10 − Узагальнена модель програмування GPU

Усі сучасні платформи API (Application Programming Interface) для програмування відеоадаптерів підтримують можливість використання декількох відеоадаптерів у межах однієї системи (наприклад, SLI у NVidia,

Crossfire/CrossfireX у AMD/ATI). Для кожного відеоадаптера можна призначити декілька контекстів виконання, і з кожним контекстом асоціюється один потік операційної системи. В рамках контексту створюються рейдери (програми для одного із ступенів графічного конвеєра, використовувані в тривимірній графіці для визначення остаточних параметрів об'єкта або зображення) та виділяється пам’ять під дані користувача (текстури у графічній інтерпретації). Одночасно на одному відеоадаптері може виконуватись лише один шейдер із одного контексту. Процеси виклику і передачі даних є асинхронними.

Способи оптимізації GPU програм

1. Маскування затримок в процесі звертання до глобальної пам’яті.

На рис. 3.11 та рис. 3.12 наведено спрощені схеми маскування затримок під час звертання до глобальної пам’яті. Потоки 1 та 2 виконуються на одному ядрі. За відсутності маскування затримок для кожного потоку відбувається така процедура [12]:

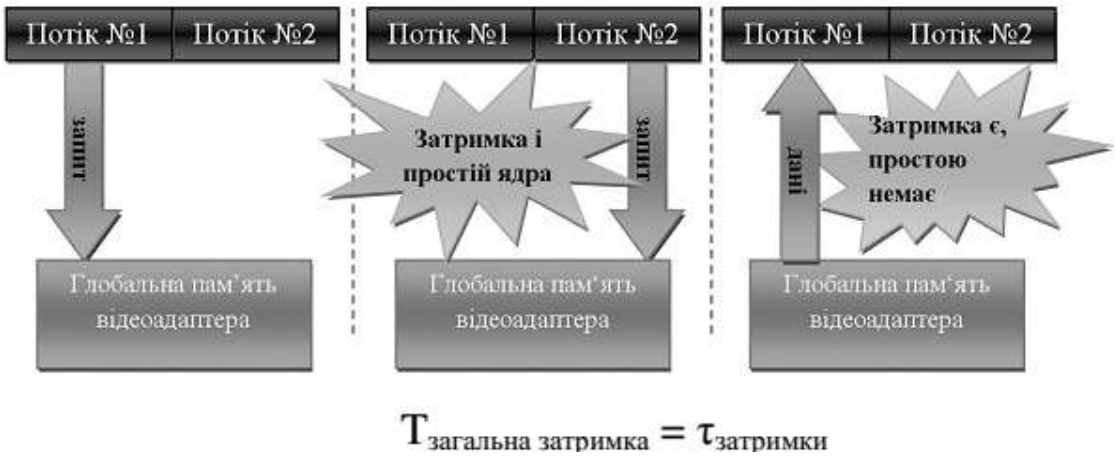
- виконується запит до пам’яті;
- ядро очікує на дані протягом 32 тактів (називається затримка або простій ядра);
- коли надходять дані із пам’яті, виконується їх обробка, і керування передається наступному потоку.

За використання схеми маскування затримок відбуваються такі дії:

- послідовно виконуються запити від всіх потоків;
- ядро очікує на дані протягом 32 тактів після першого запиту (одноразова затримка);
- дані надходять із пам’яті безперервно у порядку їх запиту, передача керування між потоками відбувається миттєво, без додаткових затримок.

$$T_{\text{загальна затримка}} = N_{\text{завантажень}} \times \tau_{\text{затримки}}$$

Рисунок 3.11 − Цикл передачі та обробки даних без маскування затримок



$$T_{\text{загальна затримка}} = \tau_{\text{затримки}}$$

Рисунок 3.12 − Цикл передачі та обробки даних з маскуванням затримок

Таким чином, за використання схеми маскування затримок максимальний час простою ядра дорівнює 32 тактам.

2. Маскування затримок під час копіювання даних з/на GPU.

Для деяких відеоадаптерів можливі асинхронні операції, тобто можливо одночасно виконувати обчислення та копіювати дані з/на відеоадаптер. Це дозволяє мінімізувати затримки під час виконання програм, що викликають виконання декількох шейдерів на відеоадаптерах. Тобто можливо одночасно виконувати один шейдер та завантажувати дані, необхідні для наступного [12].

3. Послідовне зчитування із пам'яті.

Як показують результати тестів, найкращим за часом звертання способом завантаження даних з глобальної пам'яті є послідовне зчитування даних. Одним з кращих способів організації послідовного зчитування даних є зчитування даних за адресою, яка визначається функцією від номера потоку.

4. Використання локальної пам'яті та текстурного кеша.

Під час створення алгоритму обробки даних, який потрібно виконати на відеоадаптері, потрібно намагатися поділити дані на невеликі порції, які могли б бути розміщені у локальній пам'яті і використовуватися усіма потоками, що обробляються на SIMD-блоці, та розміщувати потоки, що суміжно використовують певні дані у групах потоків для AMD StreamSDK або блоках NVidia CUDA, оскільки дані, що зчитуються сусіднім потоком, зазвичай залишаються у текстурному кеші, через який найчастіше виконується завантаження даних.

5. Узгодженість потоків в рамках одного SIMD-блока.

Для маскування затримок необхідно запускати на SIMD-блоці певну кількість потоків, при чому мінімум у 4 рази більше, ніж кількість ядер у SIMD-блоці. Кожне розгалуження алгоритму шейдера призводить до поділу потоку інструкцій на два менших потоки відповідно до гілок. Після цього ці дві групи виконуються послідовно. Такий поділ потоку інструкцій є

небажаним явищем, оскільки він призводить до неповної утилізації ресурсів відеоадаптера (не досягає максимальної ефективності), проте уникнути його можливо лише у рейдері без будь-яких розгалужень. Можливі шляхи мінімізації впливу явища поділу потоку інструкцій – це особлива організація вхідних даних залежно від групи потоків або написання розгалужень таким чином, щоб умова розгалуження залежала від номера групи потоків. Також можна запускати набагато більшу кількість потоків на SIMD-блок (ця кількість має обчислюватись для відеоадаптера, виходячи з кількості SIMD-блоків) [12].

3.4 Нейрокомп'ютери

В середині 80-х років минулого століття в США, а потім в Японії і країнах ЄЕС були розгорнуті широкомасштабні національні і міжнародні програми досліджень і розробок, які були направлені на створення нейрокомп'ютерів – обчислювальних систем на основі штучних нейронних мереж, які мають розвинений інтелект і програмуються шляхом навчання на прикладах розв'язання задач [13].

У 1995 році було завершено розробку першого нейрокомп'ютера на стандартній мікропроцесорній елементній базі.

В останні роки, через бурхливий розвиток обчислювальної техніки, теорії хаосу і теорії самоорганізації, а також на підставі досягнень синергетики і теорії дисипативних структур (структур, фазовий обсяг яких зменшується з часом) спостерігається якісний бум у розвитку *нейрокомп'ютерних технологій.*

У світі існує декілька десятків спеціалізованих фірм, які випускають продукцію в сфері нейроінформатики, крім того, багато спеціалізованих комп'ютерних фірм – IBM, Siemens Nocdorff, Mitsubisi – ведуть дослідження і мають власні розробки в цій області. Фірма Siemens в останні роки випускає спеціальні *нейрочипи.* Ці пристрої складаються з великого числа нейропроцесорів, здатних, на відміну від звичайних процесорів, робити послідовно–паралельні обчислення. Така схема обчислень пов'язана з особливістю роботи головного мозку людини, аналогом котрого і є нейрочипи.

В основу побудови нейрокомп'ютерів ліг штучний нейрон. Кожний нейрон отримує сигнали від сусідніх нейронів за допомогою спеціальних нервових волокон. Ці сигнали можуть бути збуджувальними абогальмувальними. Їх сума становить електричний потенціал у середині тіла нейрона. Коли потенціал перевищує деякий поріг, нейрон переходить у збуджений стан і посилає сигнал по вихідному нервовому волокну. Окремі штучні нейрони з'єднуються один з одним різноманітними методами. Це дозволяє створювати різноманітні нейронні мережі з різною архітектурою, правилами навчання і можливостями.

Нейрокомп'ютер – це обчислювальна система з архітектурою MSIMD (паралельно-векторна модифікація), в якій реалізовано два принципових технічних рішення: спрощено до рівня нейрона процесорний елемент однорідної структури і різко ускладнено зв'язки між елементами; програмування обчислювальної структури перенесено на зміну вагових зв'язків між процесорними елементами [13].

Загальне означення нейрокомп'ютера може бути подано таким чином.

Нейрокомп'ютер – це обчислювальна система з архітектурою апаратного і програмного забезпечення, адекватною виконанню алгоритмів, поданих у нейромережному логічному базисі.

Архітектурні особливості й апаратне забезпечення нейрокомп'ютерів

Ідея побудови автомата на основі порогових елементів, подібних до нейронів (нервових клітин), які здатні виконувати логічні функції, була сформульована більше як півстоліття тому Мак-Каллоком і Піттсом. Однак задача проектування систем на основі порогових елементів викликала великі труднощі і її рішення було знайдене лише 20 років потому. Це було настільки складним, що практично виключало можливість синтезу автоматів, які складалися більш як з десятків нейронів [13].

Системи на основі порогових елементів отримали назву *штучних нейронних мереж (ШНМ)*. Перші роботоздатні штучні нейронні системи (ШНС) були створені вже в кінці 50-х років минулого століття – перцептрон Ф. Розенблатта, система «Альфа» А. Г. Івахненко. Перші великі перцептрони на основі аналогової і цифрової техніки («Адам-А» і «Адам-Д») за межами США були створені в 1969 –71 рр. в одному з київських НДІ.

Схему перцептрона Розенблатта наведено на рис. 3.13.

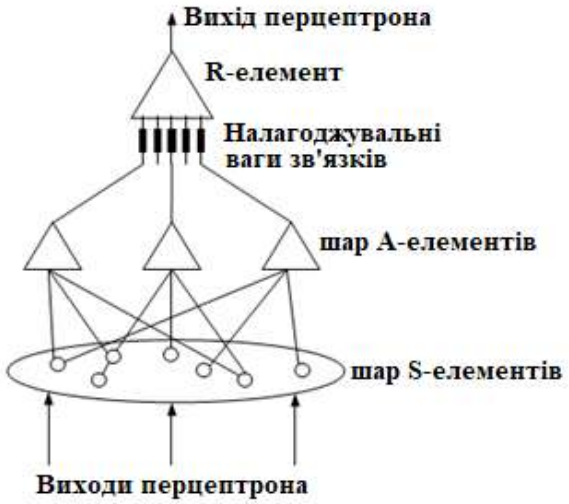


Рисунок 3.13 − Перцептрон Розенблатта

Він уміщує три шари порогових елементів. Вхідні сигнали (стимули), діючи на рецептори (S-елементи), переводять їх у збуджений стан. S-елементи випадковим чином пов'язані з сукупністю асоціативних нейронів (А-елементів). Вихід А-елемента відрізняється від нуля тільки тоді, коли збуджено достатньо велике число пов'язаних з ним рецепторів. Реакції А-елементів надходять на входи ефекторів (R-елементів) через зв'язки, ваги котрих змінюються у процесі навчання. В ефекторах обчислюється постсинаптичний потенціал – врівноважена сума сигналів, які надійшли. Як правило, в перцептроні для кожного запам'ятовувального образу виділяється один ефектор, і рішення приймається за максимальним значенням постсинаптичного потенціалу.

Властива перцептрона Розенблатта неоднорідність структури (розділення на S-, A- і R-елементи) в більш пізніх моделях ШНС втрачається.

На рис. 3.14 наведено модель штучного нейрона. Штучний нейрон імітує в першому наближенні властивості біологічного нейрона [13].

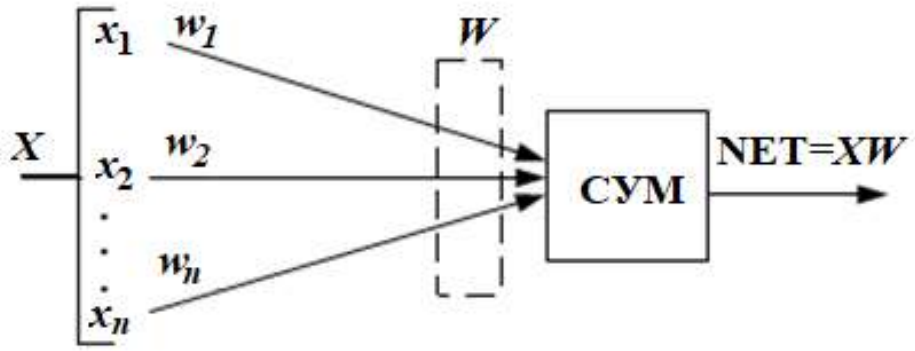


Рисунок 3.14 − Модель штучного нейрона

Множина вхідних сигналів x_1, x_2,…, x_n надходить на штучний нейрон. Ці вхідні сигнали, у сукупності позначені вектором X, відповідають сигналам, які надходять у синапси біологічного нейрона. Кожний сигнал збільшується на відповідну вагу w_i $(i = 1, 2, ..., n)$ і надходить у підсумовувальний блок СУМ (адаптивний суматор). Кожна вага відповідає «силі» одного синаптичного біологічного зв'язку. Множина ваг у сукупності утворює вектор ваг W.

Підсумовувальний блок, що відповідає тілу біологічного елемента, підсумовує зважені входи алгебраїчно, утворюючи вихід NET. У векторних позначеннях це може бути виражено таким чином: NET=XW. Сигнал NET у подальшому, як правило, перетворюється функцією активації F і дає вихідний нейронний сигнал OUT. Функція активації може бути звичайною лінійною функцією

$$OUT=K(NET),$$

де K – стала граничної функції.

OUT=1 , якщо NET>T
OUT=0 в інших випадках,

де T – деяка стала гранична величина.

На рис. 3.15 наведено структуру штучного нейрона з активаційною функцією [13].

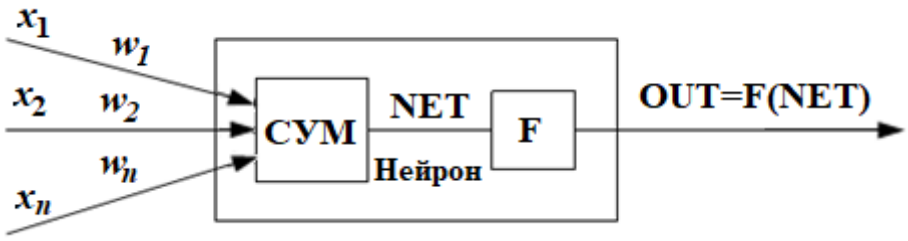


Рисунок 3.15 − Структура штучного нейрона з активаційною функцією

Блок F приймає сигнал NET і видає сигнал OUT. Якщо блок F звужує діапазон зміни величини NET так, що за будь-яких значень NET значення OUT належать деякому кінцевому інтервалу, то F називається стискальною функцією. Як стискальна функція часто використовується *логістична чи сигмоїдальна* (S-подібна) функція, що математично виражається так:

$$F(x)=\frac{1}{1+e^{-x}}.$$

Таким чином,

$$\mathrm{OUT}=\frac{1}{1+e^{-\mathrm{NET}}}.$$

За аналогією з електронними системами активаційну функцію можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Коефіцієнт підсилення обчислюється як відношення збільшення величини OUT до її невеликого збільшення, що викликало величини NET.

Розглянута модель штучного нейрона ігнорує багато властивостей свого біологічного аналога. Наприклад, вона не бере до уваги затримки в часі, що впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал.

Починаючи з перцептрона, основна увага розробників ШНС приділялась розробці і вдосконаленню методів їх навчання. Через відсутність надійної теорії навчання ці розробки носили переважно евристичний характер і отримали назву нейропарадигм.

Загалом, попри досить велику популярність нейрокомп’ютерів, ефективної комерційної стадії вони поки що не досягли. Більш популярними є *нейрокомп’ютерні* програми для персональних комп'ютерів, які призначені

для розв’язання задач апроксимації і прогнозування числових даних. Близько 5 % нейрокомп’ютерів відносяться до пристроїв професійного рівня, котрі орієнтовані на застосування потужних робочих станцій і апаратних *нейроакселераторів.* Програмне забезпечення таких систем, як правило, містить *бібліотека нейропарадигм*, що дозволяє під час розв'язання задач використовувати різні типи нейронних мереж [13].

Типовим прикладом може бути система *Brain Maker фірми CSS* (США). Система може працювати на будь-якому комп'ютері, де встановлено Windows. Базова версія орієнтована на широке коло користувачів. Її застосування не потребує спеціальних знань. Для розширення можливостей системи слугує набір додаткових програм Toolkit Option, які дозволяють прискорити процес навчання і покращити подання графічних даних.

3.5 Місце нейрокомп’ютерних технологій в науково-прикладній сфері штучного інтелекту

Основною задачею у сфері штучного інтелекту є розробка парадигм або алгоритмів, що забезпечують комп'ютерне рішення когнітивних задач, властивих людському мозку. Потрібно зауважити, що це означення штучного інтелекту не є єдиноможливим [15].

Системи штучного інтелекту мають забезпечувати рішення таких трьох класів задач: накопичення знань, застосування накопичених знань для вирішення проблеми і вилучення знань з досвіду. Системи штучного інтелекту реалізують три ключові функції: подання, міркування і навчання (рис. 3.16).

Рисунок 3.16 – Три ключові функції систем штучного інтелекту

1. *Подання*. Однією з відмінних рис систем штучного інтелекту є використання символьної мови (symbol structure) для подання спільних знань про предметну область і конкретних знань про способи вирішення задачі. Символи зазвичай формулюються в уже відомих термінах. Це робить символьне подання відносно простим і зрозумілим людині. Більш того, зрозумілість символьних систем штучного інтелекту робить їх придатними для людино-машинного спілкування.

Термін «знання», який використовується творцями систем штучного інтелекту, є всього лише ще однією назвою даних. Знання можуть мати процедурний і декларативний характер. У декларативному (declarative) поданні знання − це статичний набір фактів. Водночас існує відносно малий обсяг процедур, що використовуються для маніпуляцій цими фактами. Характерною особливістю декларативного подання є те, що в очах людини воно має зміст саме по собі, незалежно від використання в системах штучного інтелекту. У процедурному (procedural) поданні знання впроваджені в процедури, що функціонують незалежно від змісту самих знань. У більшості предметних областей потрібні одночасно обидва типи подання знань.

2. *Міркування* (reasoning). Під міркуваннями зазвичай розуміється здатність вирішувати задачі. Для того щоб систему можна було назвати розумною, вона має відповідати таким вимогам.

- Описувати і вирішувати широкий спектр задач.
- Розуміти явну (explicit) і неявну (implicit) інформацію.
- Мати механізм управління (control), що визначає операції, які виконуються для вирішення окремих задач.

Рішення задач можна розглядати як деяку задачу пошуку. В процесі пошуку використовуються правила (rules), дані (data) та керівні впливи (control). Правила діють на області даних, а керівні впливи визначаються для правил. Для прикладу розглянемо відому «задачу комівояжера». У ній потрібно знайти найкоротший маршрут з одного міста в інший. В цьому випадку всі міста, розташовані по маршруту, необхідно відвідати тільки один раз. У цій задачі безліч даних складаються з усіх можливих маршрутів та їх вартостей, поданих у формі зваженого графа. Правила визначають шляхи руху з одного міста в інше, а модуль керування вирішує, коли і які правила застосовувати.

У багатьох практичних задачах (наприклад, у медичній діагностиці) доступний набір знань є неповним або неточним. У таких ситуаціях використовуються імовірнісні міркування (probabilistic reasoning), що дозволяють системам штучного інтелекту працювати в умовах невизначеності.

3. *Навчання* (learning). У простій моделі машинного навчання (рис. 3.17) інформацію для елемента, якого навчають (learning element), надає саме середовище [15].

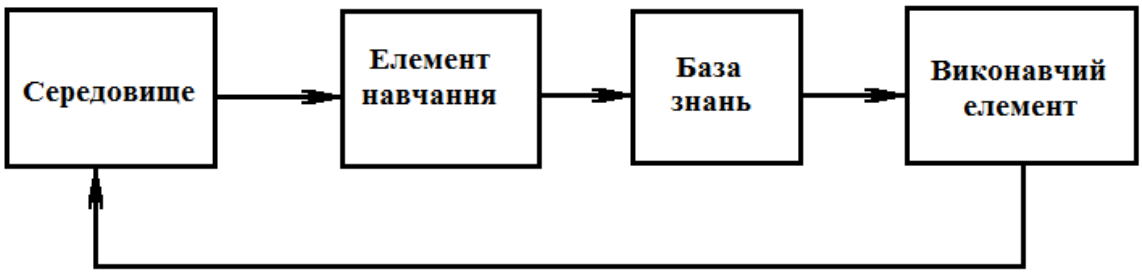


Рисунок 3.17 − Найпростіша модель машинного навчання

Елемент, якого навчають, використовує отриману інформацію для модернізації бази знань (knowledge base), знання з якої функціональний елемент (performace element) потім використовує для виконання поставленої задачі. Інформація, що надходить із зовнішнього середовища, є недосконалою, тому елемент, якого навчають, заздалегідь не знає, як заповнити прогалини або ігнорувати несуттєві деталі. Машина діє навмання, після чого отримує сигнал зворотного зв'язку (feedback) від функціонального елемента. Механізм зворотного зв'язку дозволяє системі перевіряти робочі гіпотези і переглядати їх у міру необхідності.

Машинне навчання може містити два абсолютно різних способи обробки інформації: індуктивний (inductive) і дедуктивний (deductive). За індуктивної обробки інформації загальні шаблони і правила створюються на основі практичного досвіду і потоків даних. За дедуктивної обробки інформації для визначення конкретних фактів використовуються загальні правила. Навчання на основі подібності є індуктивний процес, а доведення теорем − дедуктивний, оскільки воно спирається на відомі аксіоми і вже доведені теореми. У навчанні на основі пояснення використовується як індукція, так і дедукція.

Складності, що виникають під час навчання, і накопичений досвід привели до створення різних методів і алгоритмів поповнення баз знань. Зокрема, якщо в цій галузі працюють досвідчені професіонали, простіше отримати їх узагальнений досвід, ніж намагатися дублювати експериментальний шлях, який вони пройшли в процесі його накопичення. Цю ідею і покладено в основу експертних систем (expert system).

Виникає питання: як порівняти когнітивні моделі нейронних мереж з символьними системами штучного інтелекту? Для такого порівняння розіб'ємо проблему на три частини: рівень пояснення, стиль обробки і структуру подання [15].

1. *Рівень пояснення* (explanation level). Класичні системи штучного інтелекту (artificial intelligence − AI) основані на символьному поданні. З погляду пізнання AI передбачає існування ментального подання, в якому пізнання здійснюється як послідовна обробка (sequential processing) символьної інформації.

У центрі уваги нейронних мереж знаходяться моделі паралельної розподіленої обробки (parallel distributed processing або PDP). У цих моделях передбачається, що обробка інформації відбувається за рахунок взаємодії великої кількості нейронів, кожен з яких передає сигнали збудження і гальмування інших нейронів мережі. Більш того, в теорії нейронних мереж велика увага приділяється нейробіологічному опису процесу пізнання.

2. *Стиль обробки* (processing style). У класичних системах штучного інтелекту обробка відбувається послідовно (sequential), як і в традиційному програмуванні. Навіть якщо порядок виконання дій строго не визначений (наприклад, за сканування правил і фактів в експертних системах), операції

все одно виконуються покроково. Така послідовна обробка, найімовірніше, пояснюється послідовною особливістю природних мов і логічних висновків, а також структурою машини фон Неймана. Не можна забувати про те, що класичні системи штучного інтелекту зародилися практично в ту саму інтелектуальну еру, що і машина фон Неймана.

На відміну від них, концепція обробки інформації в нейронних мережах виникає з принципу паралелізму, який є джерелом їх гнучкості. Більш того, паралелізм може бути масовим (сотні тисяч нейронів), що надає нейронним мережам, особливу форму робастності. Якщо обчислення розподілені між безліччю нейронів, практично не важливо, що стан окремих нейронів мережі відрізняється від очікуваного. Зашумлений або неповний вхідний сигнал все одно можна розпізнати; пошкоджена мережа може продовжувати виконувати свої функції на задовільному рівні, а навчання не обов'язково має бути досконалим. Продуктивність мережі в межах деякого діапазону знижується досить повільно. Крім того, можна додатково підвищити робастність мережі, подаючи кожну властивість групою нейронів.

3. *Структура подання* (representational structure). У класичних системах штучного інтелекту як модель виступає мова мислення, тому символьне подання має квазілінгвістичну структуру. Подібно до фраз звичайної мови, вирази класичних систем, як правило, складні і складаються шляхом систематизації простих символів. З огляду на обмежену кількість символів, нові смислові вирази будуються на основі композиції символьних виразів і аналогії між синтаксичною структурою та семантикою.

З іншого боку, в нейронних мережах природа і структура подання є ключовими проблемами. У березні 1988 року в спеціальному випуску журналу Cognition були опубліковані критичні зауваження з приводу обчислювальної адекватності нейронних мереж під час вирішення когнітивних і лінгвістичних задач. Вони аргументовані тим, що нейронні мережі не задовольняють два основних критерії процесу пізнання: природу уявного подання (mental representation) і розумових процесів (mental process). Відповідно до цієї роботи наступні властивості притаманні саме системам штучного інтелекту і непритаманні нейронним мережам [15].

Уявне подання характеризується комбінаторною вибірковою структурою і комбінаторною семантикою.

Розумові процеси характеризуються чутливістю до комбінаторної структури подання, з яким вони працюють.

Таким чином, символьні моделі штучного інтелекту − це формальні системи, основані на використанні мови алгоритмів і поданні даних за принципом «зверху вниз» (top-down), а нейронні мережі − це паралельні розподілені процесори, що мають природну здатність до навчання і працюють за принципом «знизу вверх» (bottom-up). Тому під час вирішення когнітивних задач доцільно створювати структуровані моделі на основі зв'язків (structured connectionist models) або гібридні системи (hybrid

system), що об'єднують обидва підходи. Це забезпечить поєднання властивостей адаптивності, робастності і однаковості, властивих нейронним мережам, з поданнями, думками і універсальністю систем штучного інтелекту. Для реалізації цього підходу було розроблено методи отримання правил з навчених нейронних мереж. Ці результати не тільки дозволяють інтегрувати нейронні мережі з інтелектуальними машинами, а й забезпечують вирішення таких задач [15]:

- верифікація нейромережевих компонентів в програмних системах. Для цього внутрішній стан нейронної мережі перекладається в форму, зрозумілу користувачам;
- покращення узагальнювальної здатності нейронної мережі за рахунок виявлення областей вхідного простору, що не досить повно подані в навчальній множині, а також визначення умов, за яких узагальнення неможливе;
- виявлення прихованих залежностей на безлічі вхідних даних;
- інтеграція символьного і конекціоністського підходів в процесі розробки інтелектуальних машин;
- забезпечення безпеки систем, для яких вона є критичною.

3.6 Сучасні технології нейрогеймінгу

Одним із специфічних виявів комп'ютеризованого світу є нейрогеймінгові технології, які забезпечують особливий спосіб взаємодії людини з об'єктами навколишнього середовища. «На основі нейротехнологій створюються пристрої, здатні розпізнавати думки, мотиви, задуми, а також працюють над поєднанням живої та неживої матерії в єдине ціле. Це може продовжити життя пошкодженій матерії і замінити її на високотехнологічну штучну» [19].

Як відомо, існує безліч каналів, за допомогою яких людина спілкується із зовнішнім світом. По-перше, це так звані сенсорні системи (зорова, слухова, дотикальна, нюхова, смакова), які постачають людині інформацію про зовнішній світ. По-друге, це різні моторні реакції, за допомогою яких вона реагує на ці дії або реалізує власні прагнення. Травми спинного і головного мозку, церебральний параліч, м'язова дистрофія, розсіяний склероз, захворювання органів зору та слуху, численні інші захворювання можуть створювати серйозні труднощі і навіть практичну неможливість реалізації такої взаємодії. За підрахунками американських фахівців, тільки в США з такими проблемами щорічно стикаються близько 2 мільйонів чоловік (Ficke, 1991; NABMRR, 1992; Murray, Lopez, 1996; Carter, 1997), а в світі загалом – кілька десятків, а, можливо, сотень мільйонів. В цілому ряді випадків такі хворі не тільки повністю втрачають довільний м'язовий контроль рухової сфери, але навіть можуть бути повністю паралізовані. Їх життєво важливі функції підтримуються апаратними засобами, однак вони

можуть мати збережений інтелект, здатні зберегти інтерес до життя, продовжують бути коханими рідними і просто оточуючими їх людьми. Сучасні технології здатні забезпечити підтримку їх життя протягом досить тривалого часу, проте водночас виникає безліч психологічних, соціальних, економічних та інших проблем, пов'язаних з неможливістю комунікації із зовнішнім світом.

Ідея полягає в тому, щоб надати мозку хворого з моторними ушкодженнями новий, не м'язовий канал комунікації і контролю. Саме цей напрям, відомий сьогодні як Brain-computer interface (BCI) або інтерфейс мозок–комп'ютер (ІМК), пов'язаний з практичною реалізацією ідеї прямого управління безпосередньо активністю мозку тими чи іншими технічними пристроями, зі створенням каналу, здатного передавати повідомлення і команди безпосередньо від мозку до зовнішнього світу.

Нейрокомп'ютерний інтерфейс: принцип роботи

Нейрокомп'ютерний інтерфейс (НКІ або прямий нейронний інтерфейс, або мозковий інтерфейс, в англомовній літературі brain-computer interface, BCI) – фізичний інтерфейс прийому або передачі сигналів між живими нейронами біологічного організму (наприклад, мозком тварини) з одного боку, і електронним пристроєм (наприклад, комп'ютером) – з іншого боку. У односпрямованих інтерфейсів пристрої можуть або приймати сигнали від мозку, або посилати йому сигнали (наприклад, імітуючи сітківку ока в процесі відновлення зору електронним імплантантом). Двонаправлені інтерфейси дозволяють мозку і зовнішнім пристроям обмінюватися інформацією в обох напрямках. Всі існуючі технології НКІ можна розбити на два напрямки – безпосередню взаємодію з нейронами з імплантацією в тіло спеціальних пристроїв і зняття зовнішніх сигналів (переважно імпульсів мозкової активності) за допомогою зовнішніх датчиків [19].

Нейрокомп'ютерні інтерфейси, що реєструють активність нейронів за допомогою системи мікроелектродів та імплантуються безпосередньо в тканину головного мозку, мають назву інвазивні. Неінвазивні НКІ основані на методі вловлювання електричних сигналів мозку з поверхні шкіри голови, інакше кажучи, ті, що використовують електроенцефалограму.

Один з принципів роботи неінвазивних НКІ оснований на аналізі потенціалів мозку, пов'язаних з подіями. Якщо людині надати набір елементів та через 300−400 мс показати необхідний їй елемент, то виникає додатне значення потенціала, яке називається компонентом P300. Інший принцип роботи заснований на виявленні патернів. Людині пропонують, наприклад, уявляючи рух різних частин тіла, усно рахувати. Кожному виду когнітивних процесів відповідає певна амплітуда ритмів електроенцефалограми. Ці процеси розпізнаються, і кожному з них присвоюється дія. НКІ, оснований на вияві патернів, можна використовувати для керування різноманітними рухомими механізмами. Водночас існує алгоритм, що дозволяє внаслідок

тривалого навчання підсвідомо сформувати та позначити патерни, пов'язані з уявленнями про рух цього механізму, що робить керування зручним.

Нейро-комп'ютерний інтерфейс: структура

Як і будь-яка система комунікації чи управління, НКІ (рис. 3.18) має вхід, на який подається, наприклад, електрична активність мозку; модуль контролю, компоненти якого перетворюють вхідні у вихідні дані; протокол, який визначає вибір часу, початок та кінець роботи системи.

Враховуючи, що обробка сигналів в НКІ здійснюється, як правило, в цифровому форматі, обов'язковим елементом входу є аналогово-цифровий перетворювач, якщо тільки останній не є елементом другої її частини – підсилювача, на вхід якого надходять сигнали від електродів [19].

Переведені в цифрову форму сигнали піддаються різноманітним процедурам обробки (таким як просторова фільтрація, вимірювання амплітуди, спектральний аналіз, виділення єдиних нейронів тощо). Їх призначення – вилучити особливості сигналу, які можуть надійно кодувати повідомлення або команди користувача.

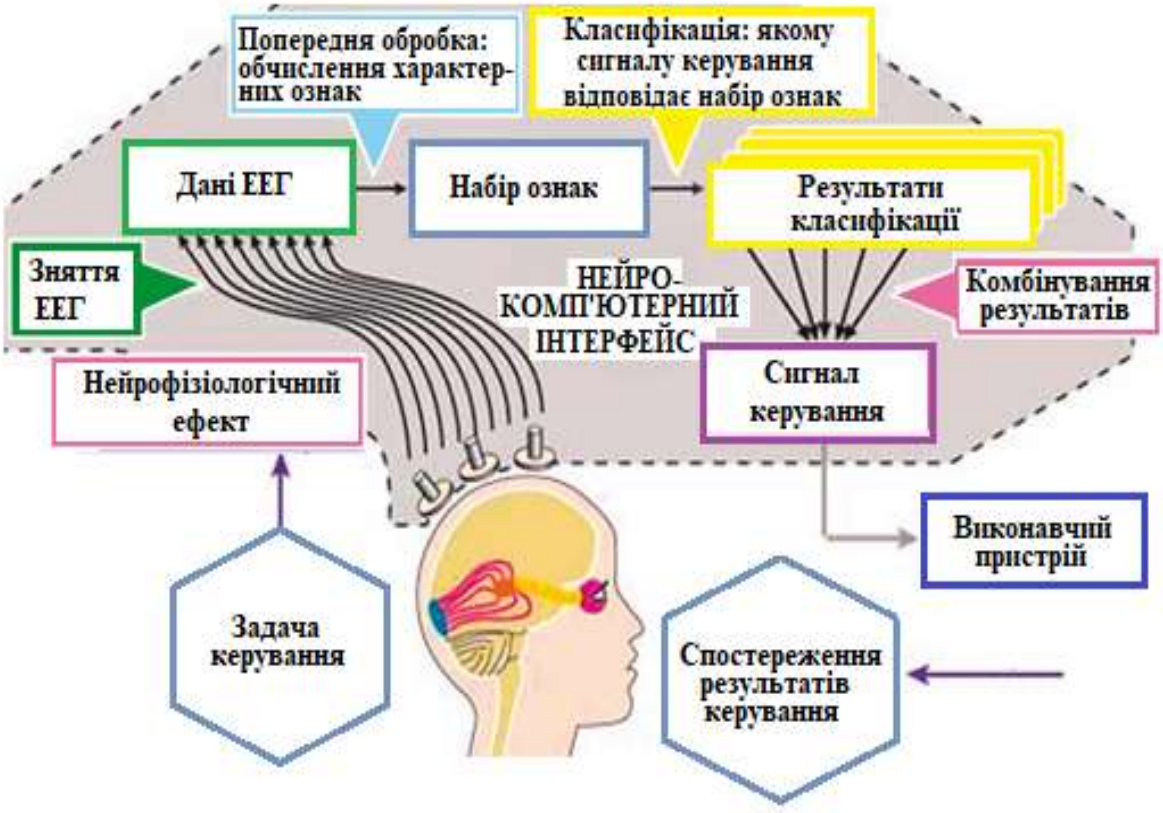


Рисунок 3.18 − Нейрокомп'ютерний інтерфейс

Одним з ключових компонентів систем НКІ є нейротехнічний інтерфейс, що складається з електродів та технічних пристроїв (підсилювачів), призначених для реєстрації електричної активності нейронів, їх груп, фокальних потенціалів або сумарної електричної активності мозку.

Цілеспрямовані зусилля всіх зацікавлених сторін та вирішення актуальних завдань нейрокомп'ютерних інтелектуальних систем зможуть забезпечити принципово нову комунікацію людини зі світом не тільки для людей з порушеннями, які послаблюють або унеможливлюють нормальну комунікацію і контроль, але також для практично здорових людей, і стати новим каналом зв'язку з цим світом, можливості якого сьогодні складно передбачити.

Просування в цьому напрямку буде залежати від багатьох чинників [19]:

- розуміння того, що НКІ та його розвиток – міждисциплінарна проблема, що потребує інтеграції зусиль нейробіологів, психологів, інженерів, математиків, інформатиків та медиків;
- вдалого пошуку способів ідентифікації сигналів (викликаних потенціалів, спонтанних ритмів, нейронної активації), які найбільш інформативні для управління і якими користувач керує найефективніше;
- розвитку навчальних методів для того, щоб допомогти користувачам поліпшити і підтримувати цей контроль;
- розробки ефективних алгоритмів для перекладу цих сигналів в команди пристрою;
- розробки точних процедур, що дозволяють оцінити ефективність застосування НКІ;
- визначення сфер застосування нейрокомп'ютерних інтелектуальних систем.

Серед зазначених проблем, мабуть, немає практично нерозв'язних навіть на сучасному технологічному рівні, що дозволяє сподіватися на появу пристроїв, які випускаються серійно та призначені для широкого кола користувачів, вже в найближчі роки.

Нейрогеймінгові технології будуть мати серйозне відображення в усіх основних сферах буття людини.

Тілесність як одна з фундаментальних сфер буття людини отримає значне розширення.

Зміни, які торкнуться тілесного і свідомого буття людини, знайдуть своє продовження в соціальній і духовній сфері, з'являться нові специфічні професії і види взаємовідносин в соціумі, нові пріоритети в духовній сфері.

Питання для самоконтролю

1. Охарактеризуйте нелінійну модель нейрона.
2. Охарактеризуйте та наведіть структури основних класів нейромережевих архітектур.
3. Наведіть порівняльну характеристику сучасних інформаційних технологій моделювання штучних нейронних мереж.
4. Розкрийте суть обчислювальних процесів у нейроподібних паралельно-ієрархічних обчислювальних системах.
5. Чому для реалізації масштабних нейронних та нейроподібних паралельно-ієрархічних мереж доцільно застосовувати GPGPU?
6. Наведіть способи оптимізації GPU програм.
7. Охарактеризуйте архітектурні особливості й апаратне забезпечення нейрокомп'ютерів.
8. Поясніть вагомість нейрокомп’ютерних технологій в науково-прикладній сфері штучного інтелекту.
9. Охарактеризуйте сучасні технології нейрогеймінгу.
10. Які чинники впливають на розвиток нейрокомп’ютерних інтелектуальних систем?

4 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

4.1 Базові компоненти СППР та розвиток структурної організації

Концепція систем підтримки прийняття рішень виникла в кінці 60-х років XX століття разом з ідеєю розподіленого комп'ютерного обчислення.

Терміна СППР (DSS) не було до 1971 року. Як уже зазначалося, термін DSS запропонували Горрі (G. Anthony Gerry) і Мортон (Michael S. Scott Morton) – професори Массачусетського технологічного інституту.

Система підтримки прийняття рішень *є інтерактивною системою, яка забезпечує користувачеві легкий доступ до моделей і даних для того, щоб підтримати процес прийняття рішень стосовно слабоструктурованих і неструктурованих завдань.*

Аналізуючи еволюцію систем підтримки прийняття рішень, можна виділити три покоління СППР: перше покоління розроблялося в період 1970÷1980 років, друге − з початку 1980 до середини 90-х років, третє − із середини 90-х років і донині (розроблення нових типів триває).

Перше покоління СППР, як уже зазначалося, майже повністю повторювало функції звичайних управлінських систем щодо надання комп'ютеризованої допомоги у прийнятті рішень.

Основні компоненти СППР мали такі ознаки (рис. 4.1) [20]:

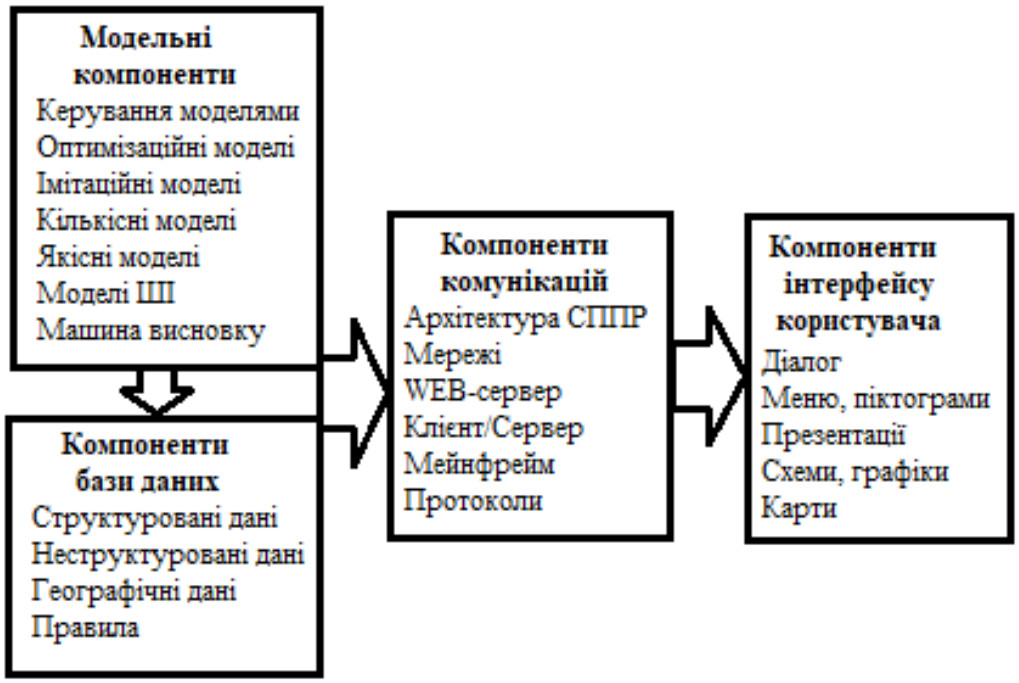


Рисунок 4.1 – Компоненти підсистем СППР

• **керування даними** − великі обсяги інформації, внутрішні і зовнішні банки даних, оброблення й оцінювання даних;

• **керування обчисленнями (моделювання)** − моделі, розроблені фахівцями в галузі інформатики для спеціальних проблем;

• **користувацький інтерфейс (мова спілкування)** − мови програмування, створені для великих ЕОМ, які використовуються виключно програмістами.

СППР другого покоління вже мали принципово нові ознаки [20]:

• **керування даними** − необхідна і достатня кількість інформації про факти згідно зі сприйняттям ОПР, що охоплює приховані допущення, інтереси та якісні оцінки;

• **керування обчисленнями і моделюванням** − гнучкі моделі, які відтворюють спосіб мислення ОПР у процесі прийняття рішень;

• **користувацький інтерфейс** − програмні засоби, «дружні» користувачеві, звичайна мова, безпосередня робота кінцевого користувача.

Дружні СППР дають їй змогу вести рівноправний зрозумілий діалог із ПЕОМ, використовуючи звичні мови спілкування. Системи можуть «персоналізувати» користувача, підстроюватися під його стиль мислення, рівень знань і професійної підготовки, а також засоби роботи.

Цілі та призначення СППР другого покоління можна визначити так:

✓ допомога в розумінні розв'язуваної проблеми. Сюди належать: структуризація проблеми, генерування постановок задач, виявлення переваг, формування критеріїв;

✓ допомога в розв'язуванні задачі: генерування і вибір моделей та методів, збір і підготовка даних, виконання обчислень, оформлення і видача результатів;

✓ допомога щодо аналізу розв'язків, тобто проведення аналізу типу «Що ..., коли ...?» та ін., пояснення ходу розв'язування, пошук і видача аналогічних рішень у минулому та їх наслідків.

СППР третього покоління мають ті самі ознаки, що і другого покоління, але з'явилися додаткові можливості за рахунок упровадження таких нових засобів інформаційних технологій та методів штучного інтелекту:

1) сховищ та вітрин даних, що дає змогу творцям рішень аналізувати величезні обсяги даних про поточні ділові транзакції з метою вибору раціонального рішення;

2) OLAP-систем, які дають можливість користувачам швидко і зручно маніпулювати великими базами даних для дослідження багатьох показників бізнесової діяльності в різних ракурсах;

3) дейтамайнінгу (Data mining) − методів інтелектуального аналізу даних для пошуку в базах і сховищах даних невідомих (прихованих) закономірностей і тенденцій;

4) консультуючих, основаних на знаннях, засобів підтримки прийняття рішень;

5) новітніх засобів телекомунікацій, які забезпечують ефективні зв'язки користувачів між собою під час створення групових рішень (Groupware), віртуальних організацій і офісів тощо;

6) географічних баз даних та геоінформаційних систем, які забезпечують користувачам доступ, показ і аналіз даних, що мають географічний (територіальний) зміст і значення, з використанням карт.

Необхідно відзначити, що до останнього часу системи підтримки прийняття рішень третього покоління інколи називали інакше, наприклад, «СППР на основі сховищ даних», «інтелектуальні СППР» тощо. Проте, як буде показано в шостому розділі, ці новітні інформаційні системи цілком уписуються в існуючі СППР як окремі типи чи групи. Крім того, не можна сказати, що нові покоління СППР витісняють чи заміняють попередні. Нині використовуються СППР усіх поколінь, можливо, у разі необхідності деякі застарілі системи модифікують.

Сучасним комп'ютерним системам підтримки прийняття рішень притаманні такі риси та властивості [20]:

1. *СППР надає керівникові допомогу в процесі прийняття рішень і забезпечує підтримку в усьому діапазоні контекстів структурованих, напівструктурованих і неструктурованих завдань*. Розум людини та інформація, що генерується комп'ютером, становлять одне ціле для прийняття рішень.

2. *СППР підтримує і посилює (але не замінює і не відміняє) міркування та оцінки керівника.* Контроль лишається за людиною. Користувач «почуває себе комфортно», використовуючи систему, завдяки зручному інтерфейсу, і не боїться працювати з нею.

3. *СППР підвищує, головно, ефективність прийнятих рішень (а не лише продуктивність ОПР).* На відміну від адміністративних інформаційних систем, в яких акцент робиться на максимальній продуктивності аналітичного процесу, у СППР значно вагомішою є ефективність процесу прийняття рішень та самих рішень.

4. *СППР інтегрує моделі та аналітичні методи зі стандартним доступом до даних і вибіркою даних*. Для надання допомоги у прийнятті рішень активізуються одна чи кілька моделей (математичних, статистичних, імітаційних, кількісних, якісних або комбінованих). Зміст баз та сховищ даних охоплює історію поточних і попередніх операцій (сильна сторона типової АІС), а також інформацію внутрішнього характеру та інформацію про середовище.

5. *СППР проста у використанні навіть для осіб, які не набули значного досвіду спілкування з ПК*. Системи є «дружніми» для користувачів, не потребують практично ніяких глибоких знань з обчислювальної техніки і забезпечують просте пересування системою, діалогову документацію, умонтовані засоби навчання та інші атрибути програмних інтерфейсних систем.

6. *СППР побудована за принципом інтерактивного розв'язування завдань*. Користувач має змогу підтримувати діалог із СППР у безперервно-

му режимі, а не обмежуватися введенням окремих команд з наступним очікуванням результатів.

7. *СППР зорієнтована на гнучкість та адаптивність для пристосування до змін у середовищі чи в підходах до розв'язування задач, які обирає користувач*. Керівник має пристосуватися до змінюваних умов сам і відповідно підготувати систему. Еволюція та адаптація системи мають бути поєднані з її життєвим циклом.

8. *СППР не мусить нав'язувати користувачеві певного процесу прийняття рішень*. Потрібно, щоб користувач мав низку можливостей обирати їх у формі та послідовності, які відповідають стилю його пізнавальної діяльності – стилю «уявлюваних моделей».

Ця традиційна характеристика систем підтримки прийняття рішень останнім часом доповнилася новими можливостями за рахунок «інтелектуалізації», зокрема:

− СППР містить модуль знань, який описує деякі аспекти світогляду творців рішень, описує, як завершити різні завдання, зазначає, які висновки мають силу за різних обставин тощо.

− СППР має здатність набувати й підтримувати дискриптивні знання (ведення записів, реєстрацію) і також інші види знань (зберігання процедур, правил тощо).

− СППР має здатність подавати знання на цей випадок (ad an hoc) у різний спосіб, а також у стандартизованих звітах.

− СППР здатна вибрати будь-яку бажану частину збережених знань для презентації або отримання нового знання засобами розпізнавання і/або розв'язування проблем.

4.2 Орієнтовані на знання СППР

Система підтримки прийняття рішень на основі знань (СППРЗ) – це інтелектуальна система, розроблена для вирішення проблем у конкретній галузі або сфері на основі знань, наданих експертами, яка містить базу знань і підтримує функції обґрунтування, пояснення та доведення. На основі обробки накопичених знань і порівняння їх з фактами, отриманими від користувача щодо конкретної проблеми, СППР може запропонувати можливо ефективне вирішення проблеми на рівні прийомів евристичної трансформації системи, надавати інтелектуальні поради, приймати рішення на рівні експерта-фахівця, а також, за бажанням користувача, пояснення процесу вирішення або причини його вибору [21].

Розрізняють два основних класи СППРЗ – експертний (ЕСППРЗ) та інтелектуальний (ІСППРЗ). Обидва ці класи належать до області ШІ. Основна відмінність між ними полягає в тому, що експертні групи зазвичай мають вузьку тематичну спрямованість і видають рішення у вигляді однієї конкретної рекомендації, а інтелектуальні не пов'язані лише з конкретною

предметною областю, а видають рішення у вигляді набору супроводжуваних прикладами варіантів рекомендацій, а остаточний вибір рішення здійснює користувач. Крім того, задачі, які розв'язуються в ІСППРЗ, менш схильні до формалізації та потребують залучення людини – користувача чи експерта – для визначення подальшого способу розв'язання задачі (вибору однієї з гілок алгоритму) на певному проміжному етапі. точки процесу прийняття рішень.

Головною відмінністю СППРЗ від інших є те, що об'єктом збирання, зберігання, обробки, передачі та використання є не дані, а знання. Знання, на відміну від даних, що відображають кількісні ознаки і подані переважно в цифровій формі, містять якісні ознаки у вигляді текстової інформації. Тому внаслідок своєї роботи годинник користувача СППРЗ отримує не документ у табличній формі, а інтелектуальну пораду у текстовій формі. Специфіка функціонування СППРЗ як інформаційного об'єкта обробки знань визначає специфіку архітектури такої системи. У загальному випадку він має вигляд, зображений на рис. 4.2, хоча окремі системи можуть не містити допоміжні блоки, наприклад, блоки мети, обґрунтування та довіри [21].

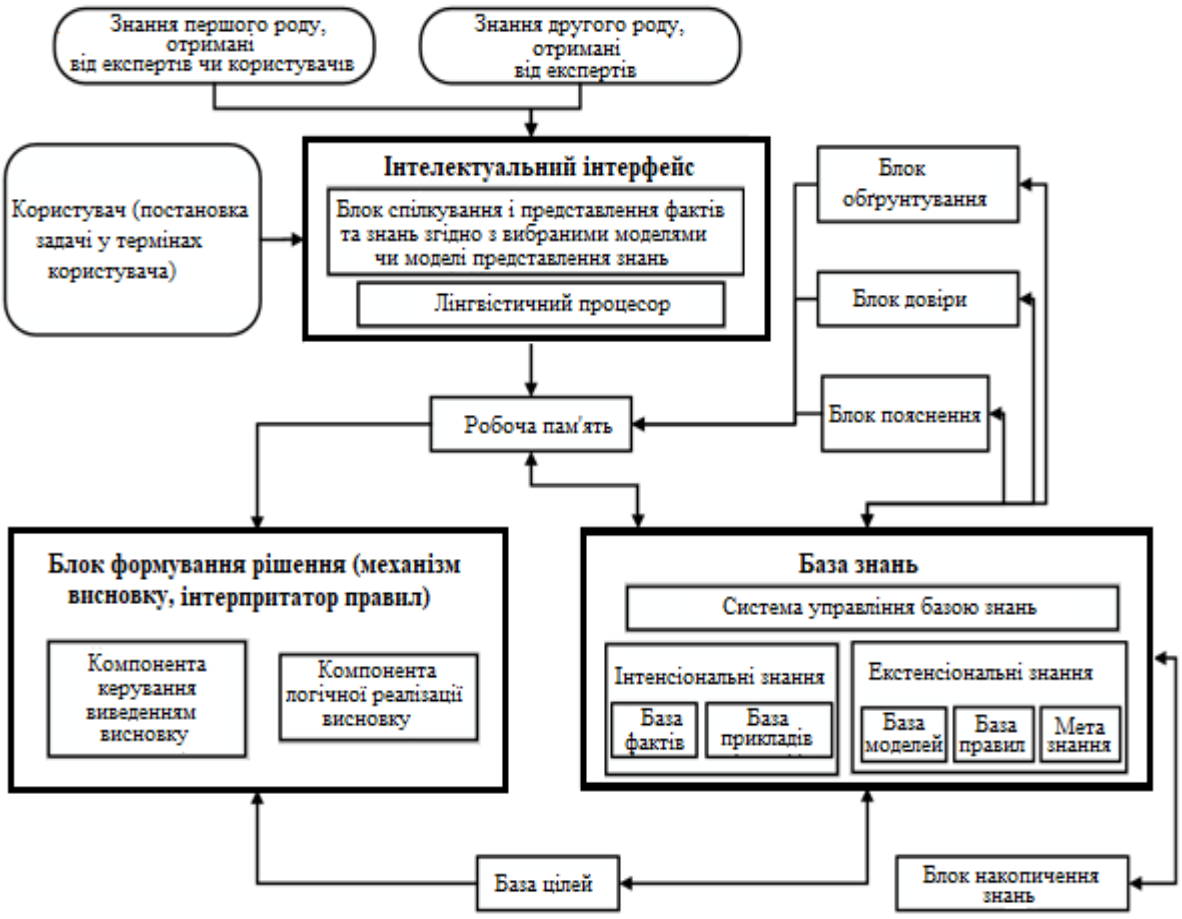


Рисунок 4.2 – Архітектура СППРЗ

База знань (БЗ) – це впорядкований набір правил, фактів, моделей, а також механізмів виведення та програмних засобів, які описують певну предметну область і призначені для подання зібраних там знань. База знань має містити як загальновідомі факти, явища, закономірності, визнані та опубліковані в цій галузі (знання І роду), так і набір емпіричних правил та інтуїтивних висновків, якими керуються фахівці, приймаючи рішення в умовах невизначеності відповідно до наявність неповної, суперечливої інформації та яка найчастіше не публікується (знання другого типу). Найчастіше в процесі обробки завдання в термінах користувача *інтелектуальний інтерфейс* у діалозі з користувачем переводить постановку завдання в терміни СППРЗ відповідно до обраної моделі подання знань і допомагає користувачеві створити модель завдання у формі, яка дозволяє порівняння з моделями в базі даних моделей бази знань. Оскільки моделі задач відповідають правилам для перетворення, оцінювання або прогнозування ситуацій з бази правил, моделювання вихідної задачі таким чином забезпечує запуск механізму логічного виведення. Результатом роботи програміста СППРЗ – ІТ-фахівця є порожня оболонка СППРЗ, в якій не заповнено базу знань. За допомогою фахівців з інженерії знань база знань доповнюється експертом – фахівцем у заданій галузі (у випадку ЕСППРЗ) або методології вирішення проблем (у випадку ІСППРЗ) – відповідно до обраної моделі подання знань, моделі задач та існуючих алгоритмів (певною мірою формалізованих). Внаслідок заповнення БЗ міститиме два типи знань – *інтенсіональні* (концептуальні, понятійні знання про об’єкти цієї предметної області та зв’язки між ними, тобто означення або опис понять через їх властивості) та *екстенсіональні* (кількісні). особливості або конкретні приклади реалізації передбачуваного знання, набір конкретних фактів, відповідних цьому поняттю). Ядром СППРЗ є база знань як модель предметної області, описана мовою, близькою до природного подання (моделювання) знань.

Основні моделі подання знань (моделі знань), які становлять набір правил для подання, опису та генерації знань у базі знань, містять, як зазначалося вище, формальні логічні моделі, рамкові моделі, продукційні моделі та семантичну мережу. У сучасних СППРЗ найчастіше використовуються останні три методи, а другий і третій є взаємодоповнювальними [21].

Крім знань, отриманих від експертів, СППРЗ містить метазнання – інформація про знання, накопичені в її базі знань, і знання про процедури, які можна виконувати з їх допомогою. Найчастіше це знання про процес вирішення задачі (контрольні знання), які використовує інтерпретатор правил; знання мови спілкування та способів організації діалогу, які використовують блок комунікації та мовний процесор; знання про способи подання та модифікації знань, що використовуються блоком накопичення знань, який може існувати незалежно або бути частиною блока прийняття рішень

у вигляді інтерпретатора знань; допоміжні структурні та управлінські знання, що використовуються в блоці пояснень.

Можливість завантажувати базу знань і редагувати накопичені в базі знання надає експерту *блок накопичення знань*. До його функцій також входить створення емпіричних залежностей від неповних знань, тобто отримання знань першого типу на основі знань другого типу. Через складність реалізації цих функцій не всі СППРЗ містять такий блок.

Управління БЗ та взаємодія з нею здійснюється на основі побудованої за тією чи іншою технологією системи управління базою знань – комплексу програмно-апаратних засобів організації та ведення бази знань.

Формулювання логічних висновків щодо конкретної проблеми на основі наявних знань, реалізація яких приводить до вирішення останніх, здійснюється за допомогою *блока формування рішень*, який ще називають *машиною (механізмом) створення висновків*, інтерпретаторами правил або просто вирішувачами задач. Цей блок керує процесом пошуку розв'язків та реалізує його разом із базою цілей, виконуючи дві основні функції [21]:

1. Перегляд наявних фактів із робочої пам'яті бази даних і вибір правил із бази знань разом із доданням (якщо можливо) нових фактів до робочої пам'яті та бази фактів.

2. Визначення порядку перегляду, способу та порядку застосування окремих правил і процедур, кількість яких у потужних СППРЗ може досягати кількох тисяч.

Крім того, в деяких ІСППРЗ цей блок виконує функцію інтерпретатора знань, реалізований як універсальна оболонка для збору, зберігання та обробки знань у вигляді символьних і графічних фреймів. Інтерпретатор знань формує знання у вигляді фреймів, а механізм логічного висновку шукає рішення через ланцюжок продукційних правил з бази правил БЗ за певним алгоритмом.

Як правило, програма механізму створення програми складається з двох компонентів: *компонент логічного виконання виведення* (який створює логічну програму на основі правил БЗ) і *компонент керування цим процесом*.

Робота компонента виведення висновку основана на правилі «*modus ponens*»: якщо відомо, що твердження А істинне й існує правило формування «ЯКЩО А, ТО Б», то твердження Б також істинне. Правила працюють, коли вони знаходять факти, які задовольняють їхню ліву сторону: істинність посилання означає істинність виведення.

Компонент керування визначає послідовність застосування правил і процедур маніпулювання знаннями та виконує чотири функції:

- порівняння – правило-приклад порівнюється з наявними фактами;
- відбір – визначення найбільш прийнятного (за заданим критерієм) серед кількох принципів, які можна застосувати в цій ситуації;

• активізація – реалізація правила у разі збігу шаблона правила з фактами або певною їх частиною;

• дія – зміна оперативної пам'яті шляхом додання до неї висновку з реалізованого правила. Якщо в правій частині правила є вказівка на дію, то ця дія виконується

Блок створення рішень (*інтерпретатор правил*) працює циклічно (рис. 4.3). У циклі він переглядає всі правила, щоб ідентифікувати ті, чиї посилання (умови, попередні) збігаються з фактами з робочої пам’яті (вони утворюють так званий конфліктний набір), і вибирає одне з них, яке є найкращим відповідно до певного критерію. Під час вибору правило активується, а висновок (факти, які формують висновок (наслідок) або зміна критерію) зберігається в робочій пам’яті. Якщо висновок містить назву дії, вона виконується. Потім цикл повторюється [21].

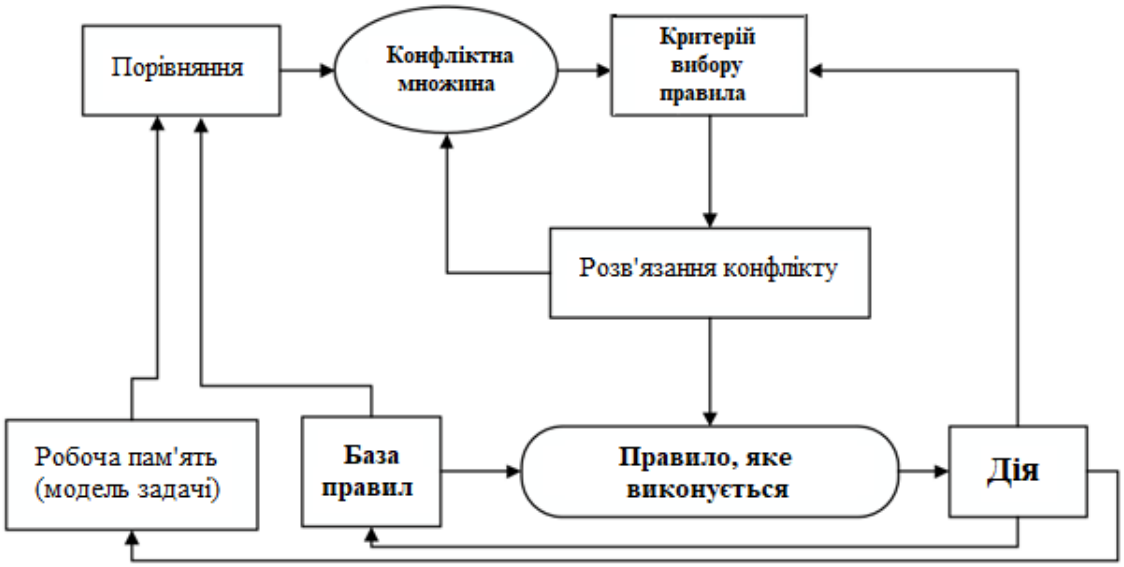


Рисунок 4.3 – Цикли функціонування інтерпретатора правил

Процес виконання дії або досягнення висновку називається *активацією правила*. Механізм логічного висновку перевіряє кожну БЗ за допомогою одного з двох методів (стратегій пошуку) – методу прямого доказу (прямий пошук, прямий вихід) і методу зворотного доказу (зворотний пошук, зворотний вихід). Відповідно до першого методу правила перевіряються одне за одним у певному, заздалегідь визначеному порядку. Активуються лише правила з *істинними* умовами в усьому ланцюжку правил. Якщо в ланцюжку є розгалуження, виконується певна кількість проходів (ітерацій), доки не будуть активовані всі можливі ланцюжки правил з істинними умовами. Їхні висновки подано користувачеві у вікні *інтелектуального інтерфейсу*, який являє собою набір програмних і апаратних засобів, що забезпечують взаємодію інтелектуальної системи з користувачем

на основі відомих понять, термінів, образів, характерних для конкретної сфери інтелектуальної діяльності людини. Основним елементом інтелектуального інтерфейсу є *лінгвістичний процесор*, який у поєднанні з комунікаційним блоком забезпечує діалог з користувачем його природною мовою та перетворення результатів цього діалогу у форму, зручну для комп'ютерної системи, тобто перетворення вхідних даних обмеженою природною мовою в дані внутрішньою мовою системи і навпаки. Під час використання методу зворотного підтвердження механізм висновку вибирає правило, припускаючи, що проблему вирішено. Рухаючись у ланцюжку правил у зворотному напрямку, вони шукають причини, які могли б підтвердити істинність цього твердження. Більшість сучасних СППРЗ здатні застосовувати різноманітні методи вирішення проблем або самостійно вибираючи найбільш підходящий для поставленої задачі, або дотримуючись інструкцій користувача, або поєднуючи обидва методи.

Робота машини створення логічного висновку залежить виключно від стану оперативної пам'яті та складу БЗ. На практиці також враховується історія, тобто робота механізму створення висновку в попередніх циклах (в деяких інтелектуальних системах окремі фрагменти відображаються в діалоговому вікні). Інформація про поведінку механізму логічного висновку зберігається в пам'яті станів, де, переважно, міститься системний протокол.

Інформація про поведінку СППРЗ за досягнення цілей у конкретній тематичній сфері міститься в компоненті інтелектуальної системи, який називається *базою цілей*. Розроблений СППРЗ також містить додаткові блоки – блок обґрунтування, блок пояснення та блок довіри, які підвищують вірогідність отримання системою дійсно ефективного рішення та полегшують його розуміння користувачем. Таким чином *блок обґрунтування* є підсистемою СППРЗ, завданням якої є перевірка відповідності отриманого рішення знанням, що містяться в базі знань; *блок пояснень* – підсистема, призначена для пояснення користувачеві методу пошуку рішення, а також самого рішення. Наявність цього блока дає можливість використовувати СППРЗ не тільки для прийняття рішень, а й як навчальну систему. *Блок довіри* – це елемент СППРЗ, призначений для підвищення рівня довіри користувачів до досягнутих результатів. Одним із способів досягнення високої довіри може бути обґрунтування – функція обґрунтування деякого рішення за участю ціннісних факторів, присутніх в інтелектуальній системі [21]. СППРЗ не мають здатності до самостійного навчання або автоматичного збору знань; нові знання вводяться в систему експертом з накопичення знань. У цьому режимі експерт вводить в систему правила (продукції) та факти щодо предметної області, в якій працює СППРЗ. Правила подано мовою, природною для користувача. Блок накопичення знань або інтерпретатор знань відповідає за підключення нових правил до бази знань. Щоб переконатися, що знання достатні (тобто процес налаштування системи

завершено), експерт перевіряє роботу системи на тестових прикладах. Якщо отриманий результат не задовольняє експерта, то за допомогою пояснювального блока він отримує інформацію про те, як саме створено цей результат і за необхідності вносить корективи у введені правила. Після завершення процесу налагодження система передається в експлуатацію користувачам.

СППРЗ, переважно, використовується для вирішення завдань, які не піддаються повній формалізації. До них, зокрема, відносяться задачі, які або не можуть бути задані в числовій формі, або мета їх розв'язування не може бути виражена у вигляді однієї, точно визначеної цільової функції, або вони взагалі не мають алгоритмічного розв'язку чи не можуть бути розв'язані. знайдено через обмежені ресурси (час, пам'ять). Водночас ІСППРЗ працює у вузьких тематичних сферах (діагностика захворювань; діагностика кредитних або страхових ризиків; будівництво складних енергоблоків, нафто- і газопроводів; оптимізація технологічних процесів, енергоменеджмент тощо), сфера застосування ІСППРЗ здебільшого обмежується не галузевими, а типовими вирішуваними проблемами. Основною сферою застосування цього типу систем є підтримка інноваційної діяльності шляхом підтримки пошуку рішень складних евристичних задач у технічних системах або бізнес-системах (сучасні системи платформи «Goldfire» («Goldfire Innovation», в попередніх версіях) «Creax Innovation Suite» компанії Creax (Великобританія), «Ideation Workbench» фірми IdeationTRIZ Corp (США), вилучення, переформатування, подання в необхідній формі та систематизація знань і фактів, що необхідні для підтримки інноваційної діяльності, з великооб'ємних інформаційних табло, поширених в корпоративних або глобальних мережах (сучасні системи платформи «Goldfire» («Goldfire Intelligence» і «Goldfire Research» в попередніх версіях (до 2002) - «CoBrain» і « Knowledgegist» корпорації Invention Machine), підсистема «Knowledgegist» системи «Creax Innovation Suite 3/1» від Creax). Найрозробленіші системи цього типу, зокрема платформа Goldfire від Invention Machine Corporation, яка об'єднує продукти лінійки Goldfire («Goldfire Innovation», «Goldfire Re-search», «Goldfire Intelligence»), є комплексними за своєю суттю та реалізовують багато різноманітних функцій управління знаннями для підтримки методичного та інформаційного супроводу вирішення завдань інноваційної діяльності.Таким чином, продукти Goldfire підтримують чотири класи методів [21]:

1. Опрацювання тексту природною мовою (вилучення знань із тексту та побудова проблемних баз даних, семантичний пошук у тексті, пошук альтернативних способів виконання або об’єктів використання певної функції, автоматичне абстрагування та анотування, автоматична класифікація документів, зокрема подання розподілу об’єктів інтелектуальної власності від певного суб’єкта за періодами, власниками, регіонами тощо; створення рецензованих оглядів, зокрема патентних документів із патентних баз да-

них у всьому світі) на цей предмет на основі інформації в корпоративній мережі та мережі Інтернет, яка забезпечує доступ до вмісту глобальної мережі, розташованої на більш ніж 2000 американських веб-сайтах урядових, академічних, дослідницьких і комерційних організацій з 26 галузей промисловості, що неможливо для звичайних пошукових систем, і їх обробка).

2. Моделювання та аналіз функціональних структур технічних об'єктів і технологічних процесів з подальшим їх удосконаленням на основі використання методів функціонально-вартісного аналізу, функціональної схеми, тримінгу(обрізки) тощо.

3. Концептуальні проекти технічних систем (технологічних об’єктів і процесів), основані на пошуку ідеї нового продукту або способу вдосконалення існуючого на основі пошуку нового фізичного способу реалізації функції продукту або усунення основних недоліків продукту-прототипу за методологією на основі теорії розвитку та розв'язування винахідницьких завдань.

4. Прогнозування розвитку технічних систем базується на еволюційних методах прогнозування на основі законів і закономірностей розвитку технічних систем. Комплексне використання таких інтелектуальних систем забезпечує підтримку вирішення найбільш типових завдань інноваційної діяльності, зокрема завдань формування інноваційної стратегії та науково-технічної політики організації, концептуального проектування технічних систем та пошуку нових ринків збуту. Їх продаж, аналіз технологій і тенденцій розвитку ринку, систематизація інтелектуальної власності організації та вдосконалення системи її управління [21].

4.3 СППР на основі OLAP-технології та сховищ даних

4.3.1 Розвиток та застосування СППР на основі сховищ даних та OLAP-систем

Системи підтримки прийняття рішень на основі сховищ даних та OLAP-систем, як і самі сховища даних (Data Warehouses) та системи аналітичного онлайнового оброблення даних, належать до типу орієнтованих на дані СППР. У загальному вигляді орієнтовану на дані систему підтримки прийняття рішень (ОДСППР) можна визначити як *інтерактивну комп’ютеризовану систему, що допомагає ОПР використовувати дуже велику базу даних із внутрішніх даних компанії і деякі зовнішні дані з навколишнього середовища системи з метою прийняття обґрунтованих рішень.* Наприклад, система може надавати дані щодо збуту продукції як самої компанії, так і її конкурентів. Деякі дані можуть бути деталізованими даними транзакцій, а деякі – агрегованими. У більшості реалізованих нині ОДСППР користувачі можуть виконувати незаплановані або в режимі на цей випадок (ad hoc) аналізи даних і формулювати запити. За допомогою таких

систем менеджери обробляють дані для ідентифікації фактів і отримання висновків у вигляді графічних зображень (діаграм, графіків, трендів) [20].

Орієнтовані на дані СППР, зокрема системи аналітичного онлайнового оброблення, інколи називають бізнес-інформаційними *(Business intelligence)*.

Саме потреба в оперативному багатоаспектному бізнес-аналізі привела до виникнення нової OLAP-технології розв'язання аналітичних завдань. Ця технологія призначена забезпечувати аналітиків динамічним багатовимірним аналізом консолідованих даних. Як уже зазначалося, розв'язання аналітичних завдань не може обмежуватись лише даними транзакційних систем. Для порівняльного аналізу та виявлення тенденцій потрібно мати великі обсяги зовнішніх даних з різних статистичних збірників, з електронних та інших джерел. Зручним способом зберігання даних для розв'язання оперативних аналітичних завдань є сховища даних, що утворюють основу аналітичних інформаційних систем. Узагальнену схему інформаційної аналітичної системи, котра ураховує описані засади, показано на рис. 4.4.

Орієнтовані на дані СППР мусять мати дані найвищої якості, інакше дані можуть призвести до невдач у розв'язанні проблем. Дані найвищої якості – це *точні*, *своєчасні*, *значимі* (важливі) і *повні* (комплектні) дані. Оцінювання або вимірювання якості джерел даних є попереднім завданням, пов'язаним з оцінюванням технічної здійснимості проекту орієнтованої на дані СППР (рис. 4.5).

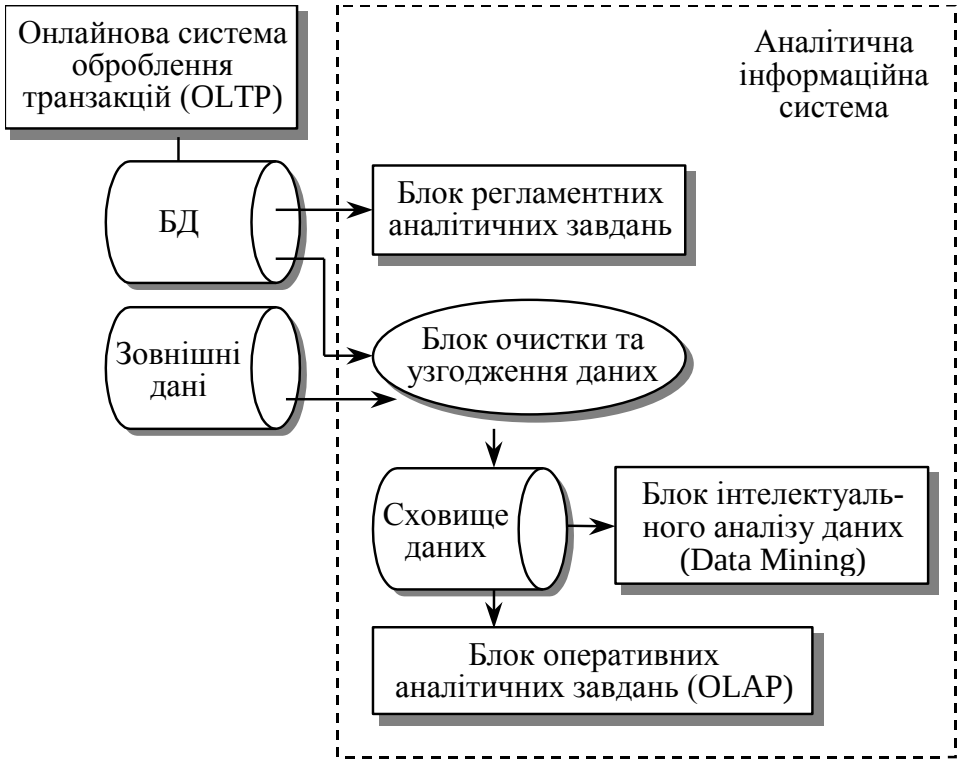


Рисунок 4.4 – Узагальнена схема інформаційної аналітичної системи

З практичного погляду OLAP є якраз тим, чого очікували від СППР протягом багатьох років, тобто перспективною системою, простою для використання, яка містить спеціалізовані (спеціально виділені) дані і пристосована до потреб користувачів. Ця система використовує сховища даних, а також містить велику кількість інструментальних засобів кінцевого користувача для організації доступу до даних і проведення їх аналізу [20].

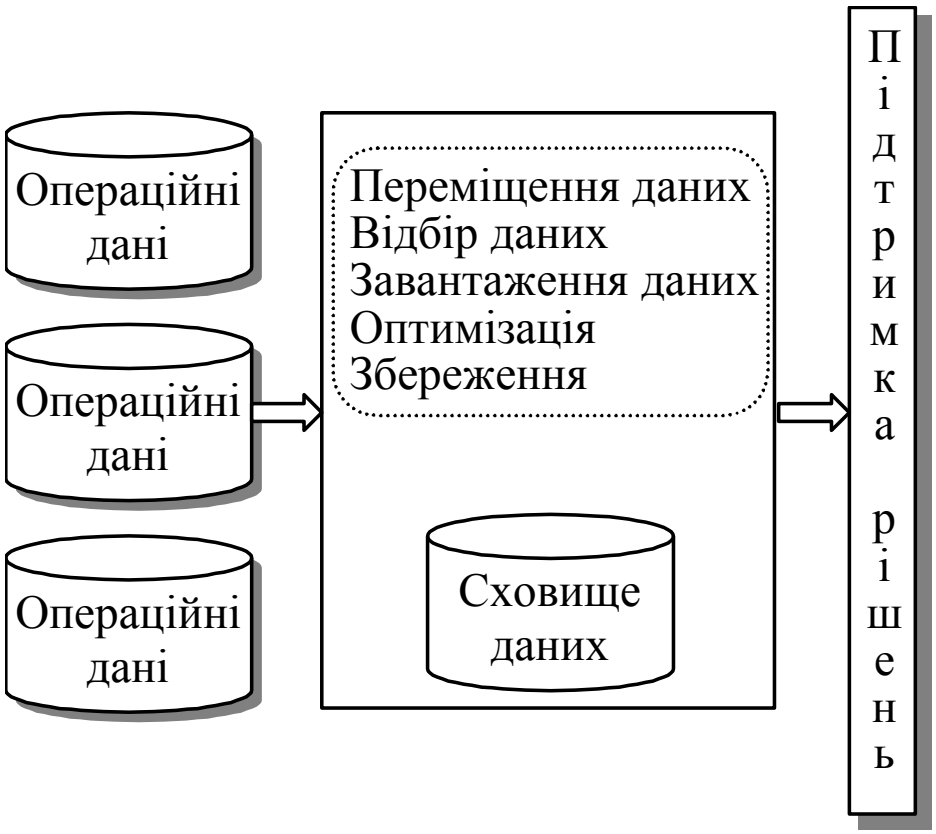


Рисунок 4.5 – Схема формування і використання сховища даних у СППР

OLAP здійснюється в багатокористувацькому клієнт/серверному режимі і уможливлює узгоджену швидку відповідь на запити, незалежно від обсягу і складності бази даних. OLAP допомагає користувачеві синтезувати інформацію підприємства завдяки порівняльному, конкретизованому перегляду даних, а також завдяки аналізу фактичних і розрахункових показників у варіантах аналізу типу «що …, якщо…?». Все це досягається за допомогою використання сервера OLAP.

4.3.2 Концепція сховищ даних і її реалізація в інформаційних системах

Практика свідчить, що різні компанії та організації будують сховища даних з метою створення своїх специфічних додатків, причому в багатьох із них перше сховище даних, зазвичай, призначається для виконавчих інформаційних систем (групових СППР). Сховище даних потенційно

орієнтовано на різні джерела, включно й операційні бази даних або дані транзакцій. Інші внутрішні дані, які надходять від електронних таблиць і внутрішньокорпоративних документів, та зовнішні дані, як наприклад бази даних новин чи цін акцій, можуть також бути збережені в сховищі даних.

У типовій організації операційні дані розпорошені через використання кількох СКБД у дуже різних форматах і на різних апаратних платформах [20].

Дані, які вибираються із різрорідних джерел, необхідно «очистити» для того, щоб перетворити їх формат на прийнятний з метою використання в сховищі. Інакше кажучи, дані мають подаватися у форматах, відповідних тим додаткам, для яких розробляється сховище даних. Наприклад, якщо деякі потрібні операційні дані рідко використовуються на індивідуальному транзакційному рівні, то їх потрібно підсумувати або агрегувати перед збереженням у сховищі.

Важливим чинником стимулювання організації сховищ даних у корпораціях стала поява спеціального програмного забезпечення, призначеного для побудови та зберігання сховища даних, а також для забезпечення доступу до них. Проте потрібно зауважити, що кількість як самих розробників, так і інструментальних засобів створення сховищ даних постійно зростає, так само як зростають обсяги продажу програмних продуктів сховищ даних, які, наприклад, ще у 2000 році перевищили 5 млрд. доларів США.

Сховище даних містить тільки моментальні знімки фактичних даних на конкретний день, як, наприклад, на останню п’ятницю або на останній день місяця. Оновлення даних залежить від потреб користувачів і бізнесового циклу, який асоціюється з даними. Наприклад, торговому менеджеру, який установлює ціни на кожний день, необхідне обновлення даних щодня або навіть частіше. Фінансовий аналіз на кінець місяця потребує тільки помісячної актуалізації даних. Технологічний цикл оновлення даних у сховищі, як правило, триває менше однієї години, тому оновлення суттєво не впливає на поточні бізнесові дії та їх результати.

Як уже зазначалося, сховище даних також містить метадані − інформацію про дані, тобто відомості про те, звідки й від кого дані надходять, хто має доступ до них і як часто, які бізнесові процеси вони забезпечують і які критичні фактори успіху вони підтримують. Ці метадані життєво важливі для підтримки сховищ даних і для кінцевих користувачів, яким потрібно знати, як розміщувати дані.

Дані, які містяться в сховищах даних, можуть бути доступними через програмне забезпечення клієнта, що використовується для підтримки прийняття рішень.

Узагальнену архітектуру показано на рис. 4.6, де виділено окремі компоненти, інструментальні засоби та джерела сховища даних.

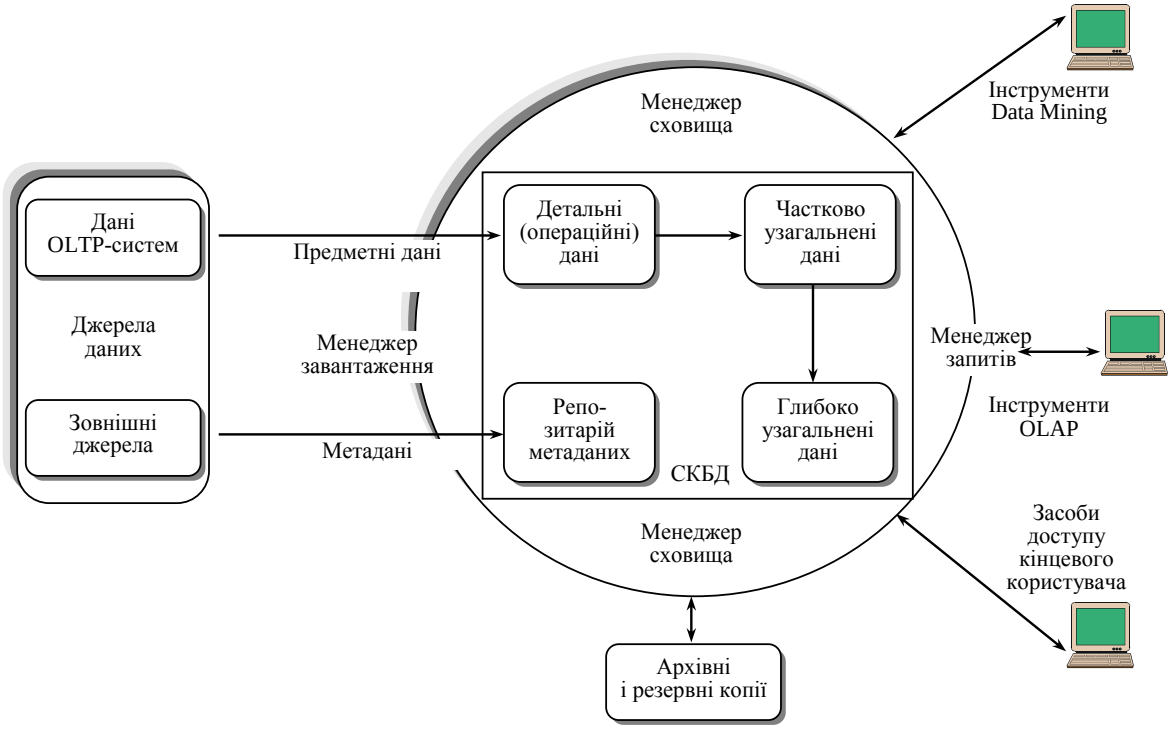


Рисунок 4.6 – Узагальнена архітектура сховища даних

Застосовуються два способи проектування сховищ даних: низхідний і висхідний. Низхідний − це такий підхід, коли спочатку проектується корпоративне сховище, а потім воно стає джерелом інформації для вітрин даних, тобто вітрини є залежними. Висхідний підхід полягає в тому, що спочатку проектуються вітрини даних, які охоплюють окремі напрями діяльності корпорації чи певні його підрозділи [20].

4.3.3 OLAP – системи аналітичного інтерактивного оброблення

OLAP (абревіатура від **On-line Analytical Processing** − інтерактивне аналітичне оброблення) фактично означає не окремі конкретні програмні продукти, а технологію багатовимірного аналізу даних, основу якої започаткувала опублікована 1993 року праця Е. Ф. Кода (E. F. Codd) «OLAP для користувачів-аналітиків: яким воно має бути», у якій він запропонував 12 правил, які виражали концепцію оперативного аналітичного оброблення даних і фактично послужили стандартом інструментальних засобів оперативного аналітичного оброблення [20].

Правила кода для OLAP

1. **Багатовимірне концептуальне зображення** (Multidimensional conceptual view). Уможливлює користувачу перегляд даних, які можна аналізувати за регіонами, часом тощо.

2. **Прозорість** (Transparency). Робить базову (що є основою) аналітичну здатність цілком прозорою для користувача.

3. **Доступність** (Accessibility). Дає змогу створювати власну логічну схему для запам’ятовування неоднорідних фізичних даних.

4. **Узгоджена продуктивність щодо створення повідомлень** (Consistent reporting performance). Забезпечує надійну продуктивність підготовки звітів для будь-якої кількості вимірів, яку обирає користувач.

5. **Архітектура клієнт/сервер** (Client/server architecture). Забезпечує мінімум зусиль для використання цієї архітектури.

6. **Генерування вимірів** (Generic dimensionality). Має тільки одну логічну структуру для подання всіх вимірів.

7. **Динамічне оброблення розріджених матриць** (Dinamic sparse matrix handling). Ефективно поводиться з пропусками в матриці, що роблять матрицю розрідженою.

8. **Багатокористувацька підтримка** (Multiuser support). Уможливлює одночасний доступ, захист і цілісність для багатьох користувачів.

9. **Необмежені перехресно-вимірні операції** (Unrestricted cross-dimensional operations). Виконує обчислення і інші операції над вимірами без втручання користувача.

10. **Інтуїтивне маніпулювання даними** (Intuitive data manipulation). Уможливлює оброблення з деталізацією (drilling down), наближення/віддалення об’єкта, переорієнтацію і консолідоване подання даних та аналізів.

11. **Гнучка організація створення звітів** (Flexible reporting). Дає змогу користувачам легко і ефективно маніпулювати звітами даних.

12. **Необмеженість вимірів і рівнів агрегації** (Unlimited dimensoins and aggregation levels). Містить щонайменше 15, а то і 20 вимірів даних.

У 1995 році до них було додано ще кілька правил, що у своїй сукупності визначили основні сучасні вимоги до OLAP-систем. Ці правила пізніше було поділено на чотири групи [20].

1. **Базові характеристики:** багатовимірність моделі даних; інтуїтивні механізми маніпулювання даними; доступність; пакетне отримання даних; клієнт-серверна архітектура; прозорість (для користувача); багатокористувацька робота.

2. **Спеціальні характеристики:** оброблення неформалізованих даних; зберігання результатів окремо від вхідних даних; виокремлення даних, яких бракує (тобто вони мусять відрізнятися від нульового значення); оброблення значень, яких бракує (всі значення, яких бракує, мають бути проігноровані в разі аналізу).

3. **Характеристики побудови звітів:** гнучкі можливості одержання звітів; стабільна продуктивність за підготовки звітів; автоматичне регулювання фізичного рівня.

4. **Керування розмірністю:** загальна функціональність; необмежена кількість вимірів і рівнів агрегування; необмежена кількість операцій над даними різних вимірів.

Ці концепції покладено в основу технології OLAP, хоча реально наявні OLAP-системи мають далеко не повний перелік описаних характеристик.

OLAP-технологія, котру можна назвати також *інтерактивним (діалоговим) аналітичним обробленням,* дає змогу на основі багатовимірної (гіперкубічної) моделі даних (на відміну від плоскої реляційної моделі даних) моделювати реальні структури і зв'язки, що є виключно важливими для аналітичних систем. Вона призначена для створення багатопараметричних моделей з метою адекватнішого відображення реальних бізнес-процесів. Технологія OLAP дає змогу швидко змінювати погляди на дані залежно від вибраних параметрів і забезпечувати особу, що приймає рішення, повною картиною щодо аналізованих ситуацій.

Усі OLAP-системи побудовані на двох базових принципах: 1) дані, необхідні для прийняття рішень, потрібно попередньо агрегувати на всіх відповідних рівнях і організувати так, щоб забезпечити максимально швидкий доступ до них; 2) мова маніпулювання даними основана на бізнес-поняттях. Дані параметруються кількома рівноправними вимірами, наприклад, дані стосовно продажу у великій торговельній компанії можна аналізувати в таких вимірах: «час» (день, тиждень, місяць, квартал, рік), «географія» (місто, область, країна), «товар» (фірма-виробник, вид товару), «покупець» (стать, вік).

4.4 Технології виконавчих інформаційних систем

До початку 80-х років двадцятого століття досить поширеною була думка, що системи підтримки прийняття рішень призначені виключно для керівників вищих і середніх ланок адміністративної ієрархії, причому вважалося, що менеджери високого рівня рідко виявляються серед кінцевих користувачів цих систем і що такі системи впливають на ефективність прийняття рішень переважно непрямим способом, тобто за рахунок використання СППР менеджерами нижчих рівнів і співробітниками, котрі обслуговують керівників високого рангу (топ-менеджерів, виконавців). Проте з розвитком інформаційних систем та завдяки підвищенню комп'ютерної грамотності виконавців багато вищих менеджерів переконалися, що прямий (on-line) доступ до організаційних даних є корисним. Ця обставина послугувала передумовою для появи нового різновиду СППР − *виконавчих інформаційних систем* (ВІС) [20].

Цей тип інформаційних систем нового покоління орієнтований на задоволення нерегламентованих (ad hoc) інформаційних потреб керівників вищого рівня; допомагає топ-виконавцям у разі необхідності проводити аналізування поточної продуктивності й коригувати заплановані дії; надає виконавцям легкий доступ до внутрішньої і зовнішньої інформації, доречної щодо критичних факторів успіху їхньої діяльності; допомагає розробляти точніше, актуальніше й цілісніше подання операцій організації, а також її конкурентів, постачальників і споживачів.

Виконавча інформаційна система − це комп'ютеризована система, яка забезпечує прямий інтерактивний (on-line) доступ до релевантної і актуальної інформації в зручному і здатному до навігації по системі форматі для підтримки створення менеджерами виконавчих рішень з використанням мережевих робочих станцій.

Релевантна і актуальна інформація − це відповідна потребам своєчасна, точна і дійова інформація про різні аспекти справ, що викликає професійний інтерес з боку вищих менеджерів. Зручність і здатність до навігації (наприклад, рух по дереву меню) системи означає, що ВІС спеціально розробляються для використання особами з обмеженим часом для обдумування проблем, недостатньою майстерністю працювати з клавіатурою і незначним досвідом роботи з комп'ютерами,

Виконавчі інформаційні системи відрізняються від традиційних інформаційних систем. Вони мають низку характерних ознак, зокрема:

- спеціально створюються для забезпечення інформаційних потреб виконавців вищого рівня і використовуються ними безпосередньо без сторонньої допомоги;
- розробляються з орієнтацією на те, що користувачі мають поверхневу комп'ютерну підготовку або не мають ніякої;
- уможливлюють доступ до даних про специфічні організаційно-управлінські питання і проблеми, а також до агрегованих, звітів;
- забезпечують користувачів багатьма оперативними (on-line) інструментальними засобами аналізу, включаючи аналіз трендів, генерування повідомлень про особливі ситуації (про відхилення) і практичне оброблення «зверху-вниз» (drill-down);
- надають можливість доступу до широкого діапазону внутрішньокорпоративних і зовнішніх джерел даних, забезпечують інтегрування інформації;
- особливо легкі для використання (за допомогою звичайної мишки або сенсорного екрана), часто настроюються на індивідуальні потреби користувачів;
- спроможні вибирати, фільтрувати, стискати і відслідковувати критичні фактори успіху або ключову індикаторну інформацію про діяльність організації.

Такі характеристики виконавчих інформаційних систем зумовили бурхливий розвиток їх різних версій розробок та високий темп упровадження. Зокрема, ще на початку 90-х років ВІС були встановлені на більше ніж половині пультів управління вищих адміністраторів найбільших компаній, а темп зростання світового продажу програмного забезпечення ВІС перевищує 18 % на рік.

Призначення ВІС. Спеціалізація ВІС − моніторинг подій і трендів, як внутрішніх, так і зовнішніх. Володіючи своєчасною і ширшою інформацією та відповідними інструментальними засобами, менеджери вищого рівня

краще готуються до прийняття стратегічних рішень» з метою створення додаткових можливостей організації і усунення проблем. Застосування виконавчих інформаційних систем може стати зброєю в боротьбі за конкурентоспроможність корпорації. ВІС часто застосовується як інструментальний засіб стратегічного планування і поліпшення якості рішень, що розробляються на верхніх рівнях управління організацією. Використання ВІС дає змогу скорочувати час, необхідний для виявлення проблем і можливостей, забезпечувати засобами для поліпшення контролю в організації, надавати швидший та ефективніший доступ до даних та моделей. Загалом можна виділити три мети розроблення ВІС [20].

Перша мета, яка бралася до уваги за створення виконавчої інформаційної системи, полягала в забезпеченні комп'ютерною підтримкою процесу навчання менеджерів стосовно регулювання загальної діяльності організації, окремих робочих процесів і взаємодії з зовнішнім середовищем. Краще підготовлені менеджери можуть краще формулювати запити до інформаційної системи розробляти ефективніші рішення.

Другою метою розроблення ВІС було забезпечення своєчасного доступу до інформації. Уся інформація, яка може міститися у ВІС, звичайно, може бути одержана менеджером традиційними методами, зокрема, шляхом створення необхідних запитів до відповідних служб або підготовкою стандартних звітів. Однак організаційні ресурси і час, які потрібні для ручного компілювання інформації в багатьох різних форматах та для відображення відповідних змін і створення специфічніших запитів, часто примушують менеджерів відмовлятися від такого способу одержання інформації, тим більше, що за час підготовки замовлених звітів стратегічні інтереси менеджерів можуть змінюватися, а тому користь від звітів, які надходять із запізненням, ніколи повністю не реалізується.

Своєчасний доступ до інформації забезпечує також значний вплив на процес управлінського навчання. Очевидно, що як тільки менеджер отримує відповідь на свій запит, то вона тут же корелюється з іншими, пов'язаними за темою запитами в голові менеджера. Якщо всі ці запити і наступні відповіді на них будуть формулюватися невідкладно, то цикл навчання відбуватиметься нерозривно. Використання традиційних методів з урахуванням тривалого часу отримання відповідей на запити користувачів призводить до того, що контекст запиту може бути втрачений, і тому цикл навчання не продовжуватиметься.

Третьою метою, з якою розробляються виконавчі інформаційні системи, є узгодження дій менеджерів та узагальнення елементів рішень. ВІС має відповідні засоби для зосередження уваги управлінців на специфічних сферах і бізнесових проблемах організації. Окремі менеджери вбачають у цьому можливість підвищувати дисципліну підлеглих, у той час як деякі підлеглі бояться директивної суті виконавчої інформаційної системи і витрачають багато часу, намагаючись перехитрити або дискредитувати її.

Жодна з цих поведінок не є відповідною або продуктивною, виходячи з загальної мети організації. Буде правильніше, якщо менеджери і їхні підлеглі працюватимуть разом для визначення докорінних причин появи складних проблем, які виявляються за допомогою ВІС.

Значний вплив на процеси організаційного управління справляють ВІС завдяки реалізованому в них принципу «вимірювання виконання». Менеджери особливо уважні до конкретної інформації щодо їх продуктивності, коли ця інформація стає доступною для їхніх начальників. Цей аспект організаційної поведінки у разі застосування ВІС може виявитися виключно корисним, якщо інформація, яка надходить до виконавців, дійсно є суттєвою і стосується мети організації. Системи, орієнтовані на подання повідомлень з необґрунтовано визначеними аспектами, можуть привести до надмірної уваги топ-менеджерів до другорядних, з погляду загальної корпоративної мети, проблем або до тих питань, які є важливими поряд з іншими, але залишеними поза увагою виконавців. Наприклад, система створення виробничих повідомлень могла б зобов'язати менеджерів зосереджуватися більше на обсягах, ніж на якості робіт.

Визначальні характеристики ВІС

Повна й ефективна реалізація призначення і цілей за проектування ВІС пов'язана з обґрунтуванням опцій щодо оброблення даних, які мають бути внесені до складу ВІС. У цьому контексті розроблено ряд базових вимог до ефективних ВІС, у табл. 4.1 наведено деякі узагальнені характеристики виконавчих інформаційних систем [20].

Найважливішою властивістю ВІС є *простота і легкість* у використанні. Добре спроектовані ВІС надають користувачеві змогу опанувати роботою з системою вже після використання обмеженої кількості команд. Крім того, в системі закладено засоби для створення зображень особливих ситуацій, автоматичного генерування відповідних звітів і аналізування трендів, що допомагає виконавцям виявляти як самі проблеми, так і можливості їх розв'язання.

По-перше, ВІС мусить мати дружній інтерфейс користувача, що заохочує його до використання системи. Часто він реалізовується за допомогою кольорових дисплеїв, з використанням мишки або сенсорних екранів.

По-друге, дані мають подаватися у форматі, легкому для розуміння, з інструментами, що дають змогу змінювати їх за необхідності, тобто ВІС має бути гнучкою щодо подання текстових, табличних і графічних зображень даних. Користуючись засобами ВІС, користувач може думати варіантно, витрачаючи для цього найменше зусиль. Така допомога містить гнучкий перегляд даних, маніпуляцію ними й режимами презентації, що підсилюють інтуїцію користувача в розумінні конкурентних тенденцій, можливостей бізнесу і пов'язаних з ним проблем. Користувач мусить мати мож-

ливість ставити запитання, пов'язані з відображенням прогнозів, стану виробничо-матеріальних запасів або планування бюджету.

Таблиця 4.1 − Інформаційні потреби особи, що приймає рішення (ОПР), реалізовані засобами ВІС

Своєчасність	Інформація має бути доступною якомога швидше. Відповідь має бути дуже короткою.
Достатність	Інформація мусить комплектною (повною). Реалізовані потреби ОПР у зовнішніх даних. ОПР мають потреби як щодо історичних, так і найновіших оперативних даних.
Рівень агрегування	Необхідно, щоб ОПР мали доступ до глобальної інформації про організацію і її конкурентів. Інформація має надаватися в єрархічному вигляді. Інформація має бути подана з різними рівнями деталізації, з можливістю уточнення «зверху вниз». Користувачам потрібні «особливі» зображення даних чи «прапорці» проблем.
Надлишковість	Має бути мінімізованою.
Зрозумілість	Система орієнтована на зберігання часу користувачів. Індикатори проблем мають висвічуватися. Мають надаватися письмові пояснення. Система має підтримувати необмежене за обсягом пояснення проблем.
Неупередженість	Інформація має бути коректною і повною. Інформація має бути затвердженою, тобто мати «законну силу».
Надійність	Доступ має бути контрольованим і надаватися тим, хто допущений до роботи в системі.
Релевантність	Система має задовольняти потреби ОПР у відповідній контексту рішення інформації.
Зіставлюваність	Потрібно, щоб ОПР мали змогу спостерігати тренди, відношення і відхилення для необхідної інтерпретації.
Відповідність формату	Гнучкість має вирішальне значення. Формат має відповідати побажанням користувача. Система має у відповідний спосіб інтегрувати текст і графіку.

Вимоги щодо моделювання у ВІС:

1. Широке використання гіпертекстових зв'язків і гіперносіїв;
2. Легкий для використання аналіз досліджуваного випадку;
3. Широке використання звітів для запобігання особливим ситуаціям і засобів відстежування причин особливих ситуацій;

4. Умовні моделі, що відбиватимуть важливі для досягнення успіху чинники;

5. Моделі прогнозування, інтегровані у всі компоненти;

6. Легка доступність щодо використання фільтрів для аналізу даних;

7. Широке використання засобів типу «що..., якщо...»;

8. Широкий ряд моделей для планування.

Вимоги щодо користувацького інтерфейсу BIC:

1. Інтерфейс має бути дружнім;

2. Інтерфейс має включати багатократно перевірений графічний інтерфейс;

3. Інтерфейс має підтримувати альтернативні пристрої введення/виведення, такі як миша, сенсорна додаткова клавіатура, сенсорні екрани тощо;

4. Система має бути доступною з кількох машин з різних місць;

5. Інтерфейс має бути інтуїтивно зрозумілим;

6. Інтерфейс має бути пристосованим до стилю керівництва індивідуального користувача;

7. Інтерфейс має містити як функцію BIC меню допомоги;

8. Інтерфейс має містити контекстно-залежне меню.

По-третє, BIC має забезпечувати якомога ширшу базу інформації. Користувач має потребу як у кількісній, так і в якісній, як у зовнішній, так і у внутрішній щодо фірми інформації. Внутрішні дані мають відбивати загально корпоративні аспекти продуктивності і операції, охоплювати як поточні, так і статистичні дані, що дають змогу проводити довгостроковий аналіз трендів. Зовнішні дані мають полегшувати оцінювання впливу на корпорацію зовнішнього середовища. BIC мусить мати добре організовані системи подання даних, що забезпечують виконавцям можливість швидко пересуватися по системі. Часто BIC пропонує «знімок» теперішнього (або минулого) стану окремих елементів бізнес-процесів у легкодоступному форматі. До того ж, система має можливості проводити уточнення (уточнювальні екрани), що надають змогу користувачеві здійснювати аналізи, які лежать в основі підсумкової інформації, щоб краще ідентифікувати проблеми / можливості. Ці уточнювальні екрани часто доповнюються можливостями виконання спеціального запиту, за допомогою якого користувач може дослідити несприятливі обставини або справи [20].

По-четверте, BIC має і швидко відгукуватися на запит, що, звичайно, потребує певного часу, необхідного системі для його опрацювання. Виконавці − це дуже зайняті особи, що звикли до швидких відповідей від підлеглих. Вони чекають не гіршого і від комп'ютерних систем. Крім того, BIC мають забезпечувати швидку реакцію користувачів на ідеї, згенеровані в самій системі. BIC має забезпечувати легкі комунікаційні можливості та

здатність генерувати звіти, що дасть змогу реагувати на надану користувачем інформацію.

Організаційно-технологічні засади створення та прийняття виконавчих рішень

Загальна характеристика виконавців. Термін «виконавець» (executive) в англомовній літературі з інформатики у зв'язку з розвитком виконавчих інформаційних систем набув значного поширення, хоча донині немає однозначного тлумачення цього терміна. Виконавці за їхнім рівнем в єрархічній структурі управління знаходяться вище менеджерів (адміністраторів), для яких, як було зазначено раніше, переважно розробляються специфічні системи підтримки прийняття рішень. Синонімами терміна «виконавець» можуть слугувати терміни «топ-менеджер», «старший менеджер» тощо. У такому розумінні в Україні частіше застосовуються терміни «керівник», «директор», «керуючий», «старший адміністратор» та ін. Значення терміна «виконавець» змінюється від однієї організації до іншої. Багато з наявних означень мають кілька спільних загальних характеристик, що допомагають зрозуміти, хто такі виконавці, що вони роблять в організаціях і чим вони відрізняються від інших менеджерів. Цими характеристиками є:

1. Виконавці керують усією організацією або її автономною частиною (наприклад, директор підприємства може розглядатися як виконавець). Їх дії орієнтовані на розвиток і процвітання підприємства загалом.

2. Виконавці відповідальні більш ніж за одну функціональну галузь організації; проте у багатьох фірмах розглядають особу, відповідальну за специфічну галузь, також виконавцем (наприклад, віце-президент з виробництва).

3. Функції виконавців, зазвичай, зосереджені на рівні стратегічного планування своїх фірм, де горизонт планування − п'ять чи більше років. Як наслідок – виконавці схильні бути більш орієнтованими на майбутнє, ніж інші менеджери.

4. Виконавці формулюють політику стратегічного розвитку організацій та визначають їх взаємодії із зовнішнім середовищем. Вони більше занепокоєні цією межею організації, ніж інші менеджери, за якими зберігається відповідальність за ефективність та ефектність внутрішніх операцій.

5. Виконавці займають дуже важливі позиції у своїх організаціях. Їх дії та рішення мають значні фінансові, трудові та бізнесові наслідки. Вони занепокоєні широким діапазоном організаційних проблем [20].

Розробляючи виконавчі інформаційні системи, потрібно враховувати персональні характеристики нинішніх виконавців, зокрема, їх вік (як правило, за 50 років), їх не досить високий рівень комп'ютерної грамотності (часто взагалі ніякої), бажання вчитися і самостійно працювати за комп'ютером тощо. Завжди доцільно з'ясовувати те, яке програмне забезпечення

та які цільові додатки використовують виконавці у своїй роботі для того, щоб бути впевненими, що віднесені до ВІС властивості не перевищують можливостей виконавців їх використовувати. Навчання виконавців щодо використання ВІС має бути короткотривалим, індивідуальним (віч-на-віч) та пристосованим до комп'ютерного обладнання виконавців.

Виконавці мають велике навантаження, їх праця пов'язана з частими перериваннями протягом дня. Робота виконавців надзвичайно складна та комплексна, виконувана за запитом, неструктурована, неспеціалізована, незосереджена, непередбачувана, неупорядкована та широкомасштабна. Робочі дії виконавців, зазвичай, короткі, різноманітні та фрагментовані й демонструють високий ступінь невизначеності. Виконавці постійно прагнуть отримувати безпосередню (з перших рук) інформацію. Вони проглядають сайти компанії, розмовляють зі службовцями та клієнтами, аналізують переважаючу тенденцію бізнесу та перевіряють легальність інформації, що надходить з різних джерел.

Виконавські завдання та функції. Усі менеджери виконують такі функції управління: планування, організації, укомплектування персоналом, розпорядження та контролю. Проте виконавці витрачають більше часу на планування та контроль, ніж інші менеджери. Головними завданнями виконавців є: стратегічне планування, управлінський та операційний контроль. Стратегічне планування стосується прийняття рішень щодо цілей організації, використання ресурсів для досягнення корпоративних цілей та управління придбанням / розміщенням ресурсів. Управлінський контроль має забезпечувати своєчасне отримання та ефективне використання ресурсів з метою досягнення цілей організації. Операційний контроль необхідний для ефективного і точного виконання специфічних завдань.

У найширшому розумінні виконавці створюють стратегічний погляд на те, що таке підприємство. Вони формулюють та визначають цілі й прагнення організації, підтримують організаційний зв'язок та несуть відповідальність за те, що необхідні для досягнення бажаних результатів заходи вжиті. Коротко діяльність виконавців зводиться до такого [20]:

- виконавці створюють і підтримують організаційну культуру. Культура організації може забезпечувати консолідоване бачення стратегій розвитку організації, створювати моральний мікроклімат для організаційного спілкування і сприяти зростанню відданості службовців своїй організації. Виконавці мають визначати стратегічні цілі та напрями розвитку фірми, розробляти генеральну стратегію, беручи до уваги культуру організації, її наявні та можливі в майбутньому ресурси і майстерність її працівників, теперішні й очікувані сприятливі або несприятливі умови;
- виконавці ведуть переговори із зацікавленими сторонами або співвласниками. Вони мають урівноважувати конкуруючі потреби і претензії трьох головних груп заінтересованих сторін: ринку капіталу, ринку проду-

ктів і організаційної клієнтури, тобто службовців усіх видів діяльності, включно й вище управління;

- виконавці планують, одержують і розподіляють трудові та інші ресурси, необхідні для досягнення стратегічної мети організації. Оскільки попит на ресурси, зазвичай, перевищує їхню наявність, виконавці мають звертати особливу увагу на розподілення ресурсів за умов їх дефіциту;
- виконавці організовують бізнес-процеси як, наприклад, планують їх, складають бюджети, контролюють їх протікання, займаються заохоченням працівників та підтримують комунікацію. Ці процеси розробляють з метою надання можливостей лінійним керівникам організовувати і керувати поточними операціями;
- виконавці відстежують і, за необхідності, управляють поточною діяльністю для забезпечення її постійного поліпшення й досягнення успіху. Вони турбуються за укомплектування штату, координацію, контроль, стандарти, виявлення «вузьких місць», розв'язання питань в арбітражному порядку та ін.;
- виконавці встановлюють регламенти, які поєднують слабко пов'язані цілі, пріоритети, стратеги і плани в русло певної сфери організаційних питань, що стосуються фінансів, маркетингу та виробництва;
- виконавці створюють оточення з осіб, які надають інформацію для розроблення і реалізації їхніх регламентів. Це оточення складається з безпосередніх підлеглих та начальників або керівників інших рівнів організації. Сюди також можна віднести постачальників, клієнтів, політиків, банкірів та інших осіб, що мають партнерські стосунки.

Виконавчі ролі. Виконання багатьох завдань, необхідних для управління складними організаціями в нинішньому мінливому середовищі потребує, щоб менеджери виконували багато різних ролей. Зокрема, Мітцберг виявив десятки їх ролей, які він згрупував у три типи: міжособові, інформаційні та вирішувальні. Усі менеджери виконують ці ролі, але виконавці мають орієнтацію, яка відрізняється від ролей у менеджерів нижчих рівнів.

Міжособові ролі складаються з ролі фігуранта (показ стороннім особам своєї організації і представлення її службовців), керівника (забезпечення керівництва і мотивації) та організатора взаємозв'язків (створення й підтримка мережі зовнішніх контактів). Як фігуранти, виконавці можуть організовувати поїздки для впливових візитерів, запрошувати важливих клієнтів на ланч або проводити церемонії відправлення службовців на пенсію. Виконавці можуть управляти безпосередньо за допомогою створення заходів мотивування і заохочення службовців (наприклад, нагородження за високу якість продукції) для досягнення організаційної мети. Як організатори взаємозв'язків, виконавці створюють контактне середовище як із зовнішнього боку, так і всередині організації, привносять свіжі і, можливо, неупереджені погляди, які можуть бути застосовані для прийняття рішень [20].

Інформаційні ролі охоплюють: роль спостерігача (бачення й інтерпретування релевантної інформації), розповсюджувача (передавання інформації до інших осіб в організації) і представника (передавання інформації до зовнішніх сторін). Як спостерігачі (монітори), виконавці переглядають інформаційне середовище і отримують за вимогою і без неї інформацію від підлеглих та інших осіб. Як розповсюджувачі вони передають цю інформацію до інших осіб як усередині, так і зовні організації.

До вирішувальних ролей належать підприємницькі новації (ініціювання та проектування керованих змін в організації), ліквідації порушень (зривів) у роботі, розподіл ресурсів та ведення переговорів. Як підприємці виконавці ініціюють трансформування організації для пристосування до частих змін у зовнішньому середовищі. Проте як ліквідатори збоїв у роботі, вони мусять іноді мати справу з неконтрольованими ситуаціями, наприклад, початком страйку. Розміщення ресурсів ставить перед ними складну проблему їх розподілу, зокрема, трудових та фінансових, на які завжди наявний попит перевищує доступну пропозицію. Виконавці також мають приймати рішення щодо розв'язання внутрішніх проблем організації.

Створення рішень виконавцями. Головне призначення виконавців − створювати рішення. Проте їхня праця – це не тільки прийняття рішень особисто, але і спостереження за тим, щоб інші особи в організації розробляли відповідні їхнім обов'язкам ефективні рішення. Мають місце деякі застереження щодо створення рішень: виконавці не мають розв'язувати питання, що є недоречними у цей момент; вони не мають приймати рішення, що є передчасними або нездійсненними.

Організаційні рішення можна поділяти на континуумі від програмованих чи структурованих (сфера дії менеджерів нижчого рівня) до непрограмованих чи неструктурованих (створюються менеджерами вищого рівня). Непрограмовані − це нові, неповторні рішення. Вони стосуються сфер, для яких альтернативи та наслідки чітко не передбачаються, а інформаційні вимоги невідомі заздалегідь. Виконавські рішення пов'язані з майбутнім і мають високий рівень неоднорідності, неоднозначності та невпевненості.

Приймаючи рішення, виконавці використовують інтуїцію та раціональний підхід для визначення проблеми, розвивають та оцінюють альтернативні напрями дій та обирають оптимальний.

Виконавська інформація. Виконавці використовують поступаючу інформацію для багатьох цілей, зокрема, щоб побачити, на що має бути направлена їхня увага, щоб виявляти організаційні проблеми, розробляти варіанти рішень і обирати напрями дій. Інформація може стимулювати творчий потенціал виконавців щодо розроблення сценаріїв розвитку подій, визначення тенденцій у зовнішньому середовищі, відстеження виконання та контролювання дій. Виконавці захищають свої інформаційні джерела [20].

Виконавці отримують інформацію як із зовнішніх, так і з внутрішніх джерел. Середовище фірми забезпечує в середньому 43 % обсягу отриму-

ваної виконавцями інформації. Інформація щодо внутрішніх дій надходить з обмеженої кількості ключових джерел. До неї належить: інформація, що міститься в стандартних робочих звітах; інформація, що отримується від підлеглих; інформація від особистих спостережень за діяльністю організації.

Зовнішня інформація охоплює зовнішнє середовище, включно й клієнтів, конкурентів, ринкові та політичні зміни, технологічні розробки тощо. Ця інформація може надходити від окремих осіб, торгових організацій та періодичних видань. Інформація відносно нових ідей та тенденцій у зовнішньому середовищі може надходити від учасників конференцій, міститися в неініційованих листах від клієнтів та замовників, у пропозиціях від постачальників, у друкованих виданнях і в мас-медіа.

Інформація може бути письмовою чи усною. Хоча письмове середовище утворює більшість виконавських інформаційних джерел, виконавці віддають перевагу усному спілкуванню через його більше інформаційне багатство. Найкорисніше усне середовище − зустрічі, зокрема, з підлеглими. З письмового середовища важливішими вважаються записки, комп'ютерні та некомп'ютерні повідомлення.

Коли виконавці стоять перед неоднозначними проблемами, вони часто покладаються на неофіційну, цілком приватну. «М'яку» інформацію, що виникає в усних комунікаціях, типу пліток, поглядів, чуток, передбачень, оцінок та пояснень.

У загальному випадку виконавці використовують отримувану інформацію для таких цілей: поширюють її серед інших менеджерів і фахівців; за її допомогою визначають і розвивають напрями дій, значимі для фірми; ідентифікують бізнесові проблеми та можливості; розвивають свої ментальні (ідеальні) моделі.

Ментальні моделі − це спрощені описи та аналогії, зображення операцій фірми, які є важливими для виконавців, інакше вони були б поглинуті великою кількістю даних. Термін «ментальна модель» започаткував 1973 року Джонсон Лагрд (Johnson Laird). На його думку, такі моделі надають можливість окремим індивідам робити висновки і передбачення з метою розуміння суті явища для вибору дії і контролю за її виконанням, а також досліджувати наступні події. Успішний виконавець аналізує елементи даних, відшукуючи в них шаблони та несумісності, і у такий спосіб синтезує інформацію у знання.

Оскільки виконавці перебувають у постійній готовності сприймати інформацію і розробляти відповідні рішення за надзвичайно складних, збурюючих умов, то ментальні моделі: дають змогу їм виокремлювати суттєві обставини, знаходити головне в зовнішньому середовищі та зменшувати невпевненість; допомагають спрощувати процес прийняття рішень, ідентифікувати важливі змінні та визначати й оцінювати альтернативи. Ефективність рішень може бути безпосередньо пов'язана з якістю ментальних моделей, що використовують виконавці. З погляду на швидкозмінювані

бізнесові умови ці моделі мають бути дуже динамічними. Їх потрібно постійно аналізувати та покращувати на основі нової інформації.

Виконавці потребують нових технологій для допомоги у формулюванні ментальних моделей, щоб ефективно обробляти великі обсяги інформації, які надходять від турбулентних внутрішнього та зовнішнього середовищ організації. Такі технології дали б змогу виконавцям займатися неоднозначними, погано визначеними завданнями. Виконавчі інформаційні системи якраз і належать до одного із типів технології, що допомагає виконавцям працювати ефективніше.

Модель та компоненти BIC

Виконавча інформаційна система як різновид СППР переважно має ті ж підсистеми, що і класична модель СППР, тобто інтерфейс користувача, засоби управління моделями, даними та повідомленнями (комунікації). Однак орієнтація цих систем на користувачів на найвищих рівнях управління вносить певні особливості в конфігурацію BIC.

Конфігурація комп'ютерних BIC, очевидно, містить персональний комп'ютер. У великих компаніях персональні комп'ютери з'єднані з мейнфреймом (центральний комп'ютер), як показано на моделі BIC на рис. 4.7 [20].

Рисунок 4.7 − Модель виконавчої інформаційної системи

Персональні комп'ютери розробників – це робочі станції. Зовнішня пам’ять – це зазвичай жорсткий диск, що містить виконувану базу даних. База даних управління містить дані та інформацію, які спочатку обробляються центральним комп’ютером компанії (мейнфреймом). Розробник вибирає параметри з меню, щоб відобразити попередньо створений екран та виконати мінімальну кількість обробки. Система також дозволяє використовувати електронну пошту компанії та доступ до зовнішніх даних у пошуку інформації. В таких випадках співробітники служби підтримки ВІС додають до наявної інформації елементи екстрених новин і пояснення.

Компоненти ВІС можна розділити на такі типи: апаратне забезпечення, програмне забезпечення, інтерфейс користувача, комунікації [20].

Обладнання. Оскільки ВІС – це інтегрована інформаційна система, яка потребує потужних пристроїв зберігання даних, тому більшість виконавчих інформаційних систем спочатку були розроблені як рішення для мейнфреймів. Дисковий простір для ВІС мав бути достатньо великим, щоб обробляти дані з усіх бізнес-секторів і враховувати майбутні розробки системи. Для розробки та обслуговування унікальної ВІС потрібен комп’ютерний персонал. Ці системи дуже дорогі, і їх використання зазвичай обмежується вищим керівництвом компанії. Як правило, використовується персональний комп’ютер з одним або декількома принтерами, щоб гарантувати взаємодію з системою найкращих розробників. Деякі розробники хочуть мати систему, яка не потребує введення з клавіатури. У цьому випадку для введення інформації використовується миша або сенсорний екран.

З появою локальних мереж (LAN) деякі варіанти ВІС стали доступними, зосереджуючись на мережевих робочих станціях. Ці системи потребують менших витрат на обслуговування та дешевшого комп’ютерного обладнання. У той самий час вони зробили інформацію ВІС доступною для багатьох користувачів у компанії. Завдяки цим перевагам з’явилась стійка тенденція переміщення виконавчих інформаційних систем від універсальних комп'ютерів до персональних комп'ютерів, підключених до мережі, і комп'ютерів середньої потужності, які виконують роль «клієнтів/серверів» мережі.

Перевагами архітектури ВІС клієнт/сервер є:

• забезпечення багаторазового подання даних, які постійно знаходяться на всіх обчислювальних платформах (від універсальних комп'ютерів до персональних) корпорації, незалежно від формату даних, тобто реляційні, ієрархічні або двовимірні файли;

• гнучка система, яка має можливість змінюватися або розширюватися залежно від змін у корпорації, реагуючи на динамічне середовище та потреби користувачів на всіх рівнях;

• можливість розробників керувати даними в режимі реального часу, забезпечуючи таким чином можливість швидкого прийняття обґрунтова-

них рішень. Зрештою, це може забезпечити компанії конкурентну перевагу, оскільки скорочує час, необхідний для проведення детального аналізу ринкової ситуації [20].

Програмне забезпечення. Програмне забезпечення для керування даними є важливим інструментом у розробці потужної BIC. Питання про те, які програмні компоненти використовувати і як вони інтегрують дані в єдину систему, підлягають ретельному вивченню.

Текстове програмне забезпечення. Багато з того, що виконавці спостерігають у своїй повсякденній діяльності, базується на тексті. Наприклад, визначення реакції конкурентів на деякі зміни в стратегічній політиці компанії потребуватиме масштабного дослідження тексту. Текст, що підтверджує цей аналіз, можна отримати як із внутрішніх, так і із зовнішніх джерел.

Одна з важливих функцій BIC – це доступ до документа за допомогою пароля для його вилучення та виконання маніпуляцій над ним. Перевага цього підходу полягає в тому, що можна знайти кілька документів, «склеїти їх разом» і зберегти як новий документ. Це дозволяє користувачеві комбінувати текстові дані, що мають відношення до прийняття правильного рішення.

База даних. Менеджерам потрібен доступ як до внутрішніх даних компанії, так і до зовнішніх даних. Дані, які надає BIC, можна отримати з кількох різних джерел. Щоб отримати конкурентну перевагу, керівникам також потрібен швидкий доступ до точної інформації. Структура бази даних визначає спосіб доступу. Переважна більшість систем використовують реляційну архітектуру бази даних. Переваги цих структур полягають у тому, що їх можна легко реалізувати або модифікувати, вони прості у використанні, і до них можна отримати доступ у різних форматах. Реляційна архітектура бази даних якраз і забезпечує гнучкість, яка особливо цінна в розподіленому середовищі або середовищі клієнт/сервер. Наприклад, підпорядкований звіт (звіт про стан) може бути заповнений під час виїзної роботи та надісланий центральному розробнику. Розробник може легко поєднати цей звіт з іншою пов’язаною інформацією (наприклад, кошторисом вартості проекту) в іншому файлі.

Графічна база даних. Для подання інформації про виконавця також можна використовувати ряд графічних інструментів: діаграми часових рядів, точкові діаграми, карти, діаграми руху, діаграми послідовності та порівняльні діаграми (гістограми). Графічне подання даних – це спосіб, який чітко передає значення та дозволяє користувачеві візуалізувати зв’язки. Графіка може перетворити великі обсяги тексту та статистичних розрахунків у зручну форму подання.

База даних моделей. Моделі – це спрощені зображення запланованих або реальних ситуацій. Моделі BIC дозволяють стандартні або спеціальні статистичні розрахунки, фінансовий та інший кількісний аналіз. Вони та-

кож прагнуть упорядкувати проблему перед її вирішенням. Моделі, які використовуються у BIC, як правило, є стратегічно орієнтованими і можуть бути оптимізаційними та неоптимізованими, щоб відображати цілі процесу прийняття рішень. Рішення, наприклад, щодо закриття заводу, найточніше моделюється за допомогою неоптимізаційних моделей. Злиття компаній або рішення щодо купівлі конкретного ресурсу, зі свого боку, можуть бути підкріплені моделлю оптимізації. Ефективна BIC має надавати користувачеві різні моделі залежно від ситуації прийняття рішення [20].

Інтерфейс. Інтерфейс користувача – це надзвичайно важливий елемент BIC, оскільки він використовується виконавцем для пошуку інформації, необхідної для прийняття рішень. Виконавчі інформаційні системи були створені, переважно, саме для цієї мети. Існує кілька типів інтерфейсів, які можуть вбудовуватися в структуру BIC. У табл. 4.2 наведено їх приклади.

Таблиця 4.2 −Типи інтерфейсів, вбудованих у BIC

Тип	Опис
Календарно заплановані звіти	Пакетно орієнтовані Попередньо визначені, підготовлені звіти Негнучкі Не потребують жодної взаємодії
Запитання/відповіді	Інтерактивні За суттю – на цей випадок (можливість «що…, якщо…,?»)
Керовані меню	Дружні до користувача Покрокові процедури Зазвичай, містять загальні, заздалегідь визначені звіти, підготовлені для користувачів
Командна мова	Попередньо визначені короткі коди мають бути вивчені користувачем
Природна мова	Як правило, англійська мова для взаємодії з BIC
Введення/виведення	Заздалегідь визначене і відоме користувачу відношення «дані/інформація»

Використовувані на цей момент BIC містять багато таких типів інтерфейсів. Ключова вимога в процесі вибору типу користувацького інтерфейсу полягає в тому, щоб інтерфейс відповідав стилістичним особливостям прийняття рішень виконавцем. Але якщо виконавця не влаштовує запропонований метод введення або виведення даних BIC, система взагалі не використовуватиметься. У майбутньому розробка інтерфейсів BIC буде зосереджена на природній мові обробки, що підвищить зручність системи.

Телекомунікації. Домінуючою тенденцією в організаційному управлінні є децентралізація операцій обробки інформації, тому телекомунікації відіграватимуть головну роль у багатьох мережевих IT-системах. Для передачі даних з одного місця в інше необхідна надійна мережа. Крім того, у конкурентному середовищі потреба у швидкому доступі до розподілених даних підвищує важливість телекомунікацій у BIC.

Подання даних у гнучких форматах, які поєднують текст, цифри та графіку, дозволяє практикам зрозуміти тенденції, які неможливо побачити лише в табличній формі. Взаємопов'язані компоненти BIC поєднують дані з багатьох джерел і дозволяють керівникам посилити контроль над бізнесом, досліджуючи зв'язки між даними.

Ефективне поєднання апаратного та програмного забезпечення, необхідного для керування системним текстом, даними, моделями та графікою, зі зручним для користувача інтерфейсом потребує постійної уваги до цих факторів. Чим більше BIC відповідає цілям і завданням компанії, тим цінішою вона буде для організації. Наявність доступних ресурсів для забезпечення такої комунікації також матиме великий вплив на успіх BIC [20].

Відмінності між BIC і СППР. Відомо, що системи підтримки прийняття рішень – це тип комп'ютерних інформаційних систем, розроблених для підтримки та вдосконалення процесу прийняття рішень. Як і BIC, СППР складається з кількох різних елементів. Хоча обидві ці системи мають можливості моделювання даних і управління, елементи подання інформації в типових СППР простіші, оскільки СППР призначені переважно для підтримки прийняття рішень керівниками середньої ланки, тоді як BIC зосереджені на підтримці прийняття рішень вищим керівництвом. Типи рішень, які приймають ці категорії виконавців, значно відрізняються.

Виконавчу інформаційну систему інколи розглядають як інтегровану систему, яка надає інформацію для створення інтелектуальних запитів користувачів, які потім можуть бути оброблені за допомогою окремого СППР. У цьому випадку аналітик, а не виконавець, виконуватиме детальний аналіз за допомогою СППР. Ця обставина пов'язана з тим, що підрядники мають знати організацію загалом, а не тільки один конкретний сектор бізнесу. СППР, як правило, надає можливість отримати дуже детальну інформацію про аналіз проблем в одній галузі бізнесу. Ще одна відмінність полягає в здатності BIC проводити попередньо запрограмоване моделювання «що, якщо…?» з самого початку.

Обидві системи відстежують і формують звіти про хід конкретних дій, але рівень деталізації обробки даних істотно відрізняється. BIC насамперед надає інтегровану інформацію. Потім його можна проаналізувати більш детально зверху вниз. СППР може надати всі деталі аналізу проблеми з першого разу. Обидва типи цих систем мають переваги та недоліки, які зведені в табл. 4.3 і табл. 4.4 [20].

Таблиця 4.3 − Переваги та недоліки виконавчої інформаційної системи

Переваги	Недоліки
Проста у використанні для виконавців вищого рівня Операції не потребують значного комп'ютерного досвіду	Обмежена функціональність Не забезпечує виконання комплексних обчислень
Надає вчасно підсумкову інформацію компанії	Важко підрахувати вигоди та виправдати застосування ВІС
Надає краще розуміння інформації	Результатна інформація може надходити багатьом «чужим» користувачам
Фільтрує дані для кращого і своєчасного управління	Система може бути занадто великою, щоб нею ефективно керувати
Надає засоби для покращення слідкування за інформацією	Важко підтримувати поточні дані Витрати на додаткове введення даних часто перевищують кошторис на це Мають місце проблеми захисту даних Може зумовити меншу надійність даних Великі витрати на створення (придбання)

Таблиця 4.4 − Переваги та недоліки СППР

Переваги	Недоліки
Проста у використанні аналітиками/технологами Націлена на нижчі рівні керівництва	Необхідні комп'ютерні навички для отримання результатів Потрібен час для підготовки, аналізу та отримання бажаної інформації
Надає інформацію та аналіз для прийняття обґрунтованих рішень	Орієнтована на деталі, занадто детально характеризує ситуації
Сприяє кращому розумінню бізнесу	Не зрозуміло, як визначати якість рішень
Покращує використання інформаційних ресурсів компанії	Важко підрахувати вигоди від СППР
Уможливлює спеціальний (на даний випадок) аналіз	Важко підтримувати цілісність баз даних
Досліджує численні альтернативи Покращує контроль та зв'язок	Надає тільки помірну підтримку зовнішніх даних та графічних можливостей

На відміну від BIC, використання СППР має визначатися на основі потреб окремої ситуації. BIC призначена для забезпечення дуже високого рівня агрегування інформації про корпоративну діяльність. Це також дозволяє отримати додаткові пояснення щодо даних, які використовувалися для створення звітів. У міру збільшення розмірів корпорацій загроза комунікаційного перевантаження, викликана необхідністю забезпечення ефективного використання BIC, стає реальною.

Потік інформації утворюється внаслідок діяльності багатьох підлеглих одного виконавця, тому є загроза його перевантаження тими даними, що надходять від них. Для запобігання такого стану BIC потрібно розробляти тільки для пошуку даних, які стосуються поточної ситуації або проблеми. Вимоги підтримки зовнішніх та внутрішніх баз даних, змінні та зв'язки, що є елементами моделей BIC, можуть створювати проблеми цілісності даних. Персонал компанії, котрий обслуговує BIC, може потребувати додаткових ресурсів для виконання експлуатаційних операцій.

4.5 Впровадження та оцінювання СППР

Реалізація СППР означає впровадження планової системи. Реалізація передбачає перетворення проекту в код, але це вже далеко виходить за межі програмування. Такий підхід передбачає створення та попереднє завантаження бази даних та бази моделі, керування кінцевим продуктом, що містить встановлення (інсталяцію), введення в експлуатацію, компонування та фактичне тестування. Ще одним аспектом упровадження СППР є навчання користувачів та забезпечення того, щоб вони сприймали СППР як корисний та надійний інструментальний засіб. І нарешті, оцінювання містить всі ті кроки, які б гарантували, що система здійснює те, що потрібно, і виконує все добре. Стисло розглянемо питання впровадження та оцінювання СППР [20].

Стратегії впровадження

Успіх кожного впровадження значною мірою залежить від процесу, прийнятого командою впроваджувальників. Немає стандартних кроків для забезпечення успіху впровадження СППР: підхід, який добре реалізовано в одній ситуації, може бути непридатним в іншій ситуації. У 1988 році Свенсон (Swanson) виділив 9 ключових факторів успіху або невдачі інформаційних систем, які також містять і СППР. Вони стосуються оцінення як самої системи (якість розробки та рівень впровадження), так і процесу розробки (залучення користувачів, взаєморозуміння та управління проектом), а також організації, в якій буде використовуватися СППР (наприклад, обов’язки керівництва, відповідність ресурсів ситуаційній стабільності).

Досягнення добрих кондицій СППР

Добрі кондиції СППР є гарантією того, що система буде виконувати саме те, що від неї очікується. Успіх впровадження СППР багато в чому

залежить від якості системи, простоти та гнучкості її використання. Зрозуміло, що якщо ОПР не знають, що СППР допомагає їм приймати рішення, вони не використовуватимуть це.

Найбільшою допомогою, яку може надати система, є організація доступу до інформації, про яку організатор може не знати, надання прикладів, яких організатор може не мати, та збір інформації, яка в іншому випадку була б ізольованою для більш відповідного використання ОПР. Крім того, чим легший доступ до інформації та моделей, тим краще вони будуть використовуватися ОПР. Ключовими факторами успішного вирішення цього кола проблем є використання прототипів та опитування користувачів. Ці питання обговорювалися раніше.

Додержуватися простого розв'язання

Важливо, щоб СППР надавала саме ту підтримку, яку очікують користувачі. Це означає, що система має надавати інструменти, необхідні для створення рішень без використання складних технологій, освоєння яких потребує від користувача великих зусиль. Дуже часто проектувальники втрачають бачення потреб користувачів і намагаються замість цього забезпечити їх останніми «ноу-хау» або всіма «застереженнями», пов'язаними з доступною технологією. Або проектувальники комп'ютеризують частину операцій тому, що це можна зробити, а не тому, що це полегшить процес створення рішень. Звичайно, на прохання користувачів розробники можуть надати можливість поекспериментувати з цими технологіями, але така ситуація здається лише відхиленням для отримання «справжньої роботи» для ОПР. Тому такі методи можуть затягнути процес впровадження.

Більшість потреб у прийнятті рішень не є «простими». У цьому випадку СППР неможливо розробити простим способом. Однак система має бути простою з погляду потреб ОПР. Як правило, користувачам не потрібно точно знати, як саме система виконує операції [20].

Оцінювання впровадження системи

Питання про те, як проектувальник може знати, коли СПД і її впровадження буде успішним, є досить складним і неоднозначним. По суті, воно зводиться до вибору параметрів і методики оцінювання СППР. Як уже зазначалось раніше, оцінювання СППР має проводитися на всіх етапах її розроблення. Проте найскладніше оцінити успішність саме процесу впровадження системи. Зокрема, ця оцінка має визначити, як СПД допоможе організації отримати додаткові ресурси та як це сприятиме покращенню використання обмежених ресурсів; як СППР вплине на ефективність організації загалом завдяки його впровадженню тощо. СППР можна оцінити з погляду початкових витрат або вигод для організації, але жоден із цих двох факторів не допомагає розробникам у вдосконаленні системи, особливо тому, що, як уже було доведено, провести техніко-економічний аналіз аналіз проекту СППР практично неможливо.

Загалом СППР має допомогти визначити інформаційні потреби ОПР, бути простою у використанні, надавати можливості для дослідження, надавати інтелектуальну підтримку та легко виконувати всі свої функції. Як послуга, призначена для підтримки прийняття рішень, ця система також має відповідати потребам прийняття рішень, відповідати організаційним обмеженням і бути прийнятною для користувачів. Тому, щоб реалізація була успішною, розробник має спрямувати зусилля на технічну та організаційну придатність СПД.

Технічна відповідність (придатність)

Якщо СППР не задовольняє технічні вимоги і необхідні функції користувачів, то така система використовуватись не буде. Якщо система не використовується, тоді впровадження буде невдалим. Отже, одним із можливих вимірів визначення успіху впровадження є рівень використання СППР, особливо порівняно з потребами користувачів. Водночас більш прагматичною може бути оцінка ряду ознак, що відповідають інформаційним потребам користувачів, особливо порівняно з набором можливих властивостей інформації на виході СППР (актуальність, достатність, рівень деталізації та агрегування, резервування (надлишковість), зрозумілість, незалежність від упередження, надійність, релевантність для рішень, ефективність витрат на інформацію, порівнюваність, можливість квантифікації, відповідність формату). Якщо система забезпечує користувача інформацією, яка є сумісною з потребами підтримки прийняття рішень за всіма вимогами до вхідної інформації, то вона матиме успіх.

СППР має забезпечувати потреби у зміні моделей і щодо певних операцій для керування ними, таких, наприклад, як інтелектуальна допомога та інтеграція моделей. Якщо СППР забезпечує відповідні зміни моделей та можливості керування ними, то вона буде вважатися вдалою [20].

Організаційна відповідність (придатність)

Організаційна сумісність СППР може означати, що система має стати компонентом загальної системи організації. Це потребує підтримки стилів прийняття рішень користувачами та способів, якими ці стилі прийняття рішень змінюються з часом. Крім того, існує необхідність відповідності організації, в якій діє СППР. Він має забезпечувати певний рівень захисту інформації та використовуватися відповідно до корпоративної політики, а також надавати інформацію відповідно до очікувань користувачів. Так само, як нові співробітники мають адаптуватися до відділу та організації, СППР має зробити те саме. Це може означати відповідність інтерфейсу користувача, відповідність готовності даних, відповідність методологій моделювання стилю прийняття рішень в організації. Якщо система не може пристосуватися до певного відділу, то вона буде «страждати» так само, як

новий працівник організації, який не зміг адаптуватися, а отже, така СППР впроваджена не буде.

Зокрема, адекватність реалізації можна оцінити шляхом визначення управлінських характеристик системи; якою мірою задовольняються інформаційні потреби менеджерів; вплив проекту на комп'ютерну діяльність компанії. Ці рейтинги відображають наше сприйняття системи. Крім того, всі вони визначаються після впровадження системи. Тому їх не можна використовувати для планування та реалізації всього проекту. Кращим методом є оцінення різних типів нетехнічних здібностей. СППР також потрібно коригувати в необхідних місцях, визначених організацією [20].

Питання для самоконтролю

1. Наведіть базові компоненти систем підтримки прийняття рішень та їх основні властивості.
2. Охарактеризуйте орієнтовані на знання СППР (СППРЗ).
3. Наведіть та охарактеризуйте два основних класи СППРЗ.
4. Охарактеризуйте СППР на основі OLAP-технології та сховищ даних.
5. Технології виконавчих інформаційних систем (ВІС): призначення та їх характеристики.
6. Розкрийте суть ВІС та охарактеризуйте функції виконавців.
7. Наведіть та охарактеризуйте модель виконавчої інформаційної системи.
8. Наведіть та охарактеризуйте компоненти ВІС.
9. Охарактеризуйте відмінності між ВІС та СППР.
10. Розкрийте суть впровадження та оцінювання СППР.

5 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЕКСПЕРТНИХ СИСТЕМ

5.1 Взаємозв'язок експертних систем та систем штучного інтелекту. Поняття експертного аналізу. Основні характеристики експертних систем

Експертні системи є відносно молодою галуззю науки, яка була створена в рамках досліджень штучного інтелекту в середині 1970-х років. Своїм розвитком вони зобов'язані значним змінам, що відбулися на той час у технології створення та використання програмного забезпечення з використанням штучного інтелекту. Найважливішими з них є [22, 23]:

- відокремлення універсальної частини програми (логічного виводу) від частини, залежної від тематичної області (бази знань);
- підвищення рівня взаємодії користувача з комп'ютерною програмою.

Виконання таких умов дозволяє віднести комп'ютерну програму до класу експертних систем:

• програма має ґрунтуватися на знаннях. Досить виконати певний алгоритм, наприклад, розібрати об'єкти зі списку на наявність певної властивості, яка явно не відповідає таким вимогам. Наприклад, це еквівалентно тому, щоб надати випадковій людині список симптомів і відповідних ліків та очікувати, що вона зможе ефективно лікувати людей. Дуже скоро вона зіткнеться з ситуацією, яку не передбачила в отриманому списку;

• знання, які використовує система, мають бути зосереджені в певній тематичній області. Випадковий набір імен, дат, місць подій тощо – не є знаннями, які є основою для експертних аналізів. Знання потребують певної організації та інтеграції, тобто окрема інформація має бути пов'язана між собою і створювати щось на зразок ланцюга, в якому одна ланка безпосередньо пов'язана з іншою;

• рішення проблеми має випливати безпосередньо з наявних у системі знань. Проста демонстрація певних знань, наприклад, як обслуговувати комп'ютер, не означає, що його можна відремонтувати. Також отримати доступ до технічної документації не означає запросити спеціаліста, який зможе вирішити проблему.

Таким чином, можна дати загальне означення експертних систем, яке підтверджує вчений Фейгенбаум: «Експертна система – це інтелектуальна комп'ютерна програма, яка використовує знання та процедури виведення для розв'язання проблем у певній проблемній області, настільки складних, що для їх розв'язання потрібно запрошувати експерта, а також виробляє рекомендації для розв'язання цих проблем».

Системи, що основані на знаннях, зберігають свої правила розв'язання проблем конкретної проблемної області у базах знань. Проблема подається

системі у вигляді набору фактів, що описують певну ситуацію. У той самий час система використовує базу знань, щоб спробувати зробити висновки з цих фактів.

Щоб пояснити важливість *експертного аналізу*, розглянемо простий приклад. Після встановлення певної нової програми вона не працює, але користувач отримує повідомлення на кшталт «Call to Undefined Link». Як і більшість інших повідомлень, це йому мало допоможе. Він видаляє весь каталог і перевстановлює програму, але результат не змінюється. Він змінює налаштування в різних файлах ініціалізації, але це теж не допомагає. І тоді користувач звертається до людини, яка працює в сервісному центрі і займається цими проблемами. Він радить видалити деякі застарілі модулі DLL із системного каталогу та перевстановити програму. Виконуючи його вказівки, користувач досягнете успіху протягом десяти хвилин.

Таким чином, незалежно від рівня *експертного аналізу*, необхідного в цій галузі, фахівець з обслуговування може це зробити, а людина, навіть з достатніми знаннями в галузі інформатики, яка програмує завдання тільки в межах своєї галузі – він не може це зробити, тому що не має досвіду у вирішенні таких проблем. Тому вміння проводити експертний аналіз – це не лише питання наявності певних знань і рівня навичок. Для цього необхідно мати дуже специфічні навички та вміння розібратися в конкретній ситуації в межах цієї сфери. Це означає, що бути фахівцем і мати загальну освіту – не одне й те саме.

Предметом теорії ЕС є методи та техніки для проектування людино-машинних систем, компетентних у вузькоспеціалізованій галузі. Така компетентність передбачає знання теми, розуміння проблем у цій сфері та здатність вирішувати деякі з цих проблем. Знання, які використовуються в будь-якій галузі, як правило, бувають двох типів: загальнодоступні (факти, визначення, теорії, викладені в довідниках, практикумах і підручниках) та індивідуальні (які ґрунтуються на особистому досвіді фахівця і є більше, ніж загальновідомі). Таке індивідуальне знання складається переважно з емпіричних правил, які називаються евристикою. Саме евристика дозволяє фахівцям знаходити ефективні способи вирішення проблем в умовах викривлених і неповних даних.

Відмінною рисою ЕС є можливість акумулювати знання та досвід найдосвідченіших фахівців у певній вузькій тематичній галузі, завдяки чому користувачі ЕС із середніми навичками можуть вирішувати поточні проблеми так само ефективно, як і експерт. Такий ефект досягається завдяки тому, що в своїй роботі ЕС використовує ту саму схему аргументації, яку використовував би експерт у тій чи іншій ситуації. Таким чином, це дозволяє зберігати, збирати та поширювати знання, роблячи унікальний досвід кількох висококласних спеціалістів доступним для широкої групи звичайних фахівців [22, 23].

Можна навести такі основні характеристики, що відрізняють ЕС від звичайних програм [22, 23]:

- ЕС моделює не стільки фізичну або якусь іншу природу певної проблемної області, скільки механізм мислення людини стосовно розв'язання задач з цієї області. Це суттєво відрізняє ЕС від систем математичного моделювання чи комп'ютерної анімації. Не можна однозначно стверджувати, що програма повністю відтворює психологічну модель фахівця, але основна увага приділяється комп'ютерному поданню методики вирішення проблем, яку використовує експерт;
- ЕС, крім виконання певних обчислювальних операцій, формулює певні міркування та висновки на основі знань, які використовує. Знання в ЕС зазвичай подано якоюсь спеціальною мовою та зберігаються окремо від самого програмного коду, який формулює висновки та міркування. Цей компонент називається базою знань;
- під час розв'язування задачі домінують евристичні та наближені методи, які, на відміну від алгоритмічних, не завжди гарантують успіх. Евристики створюються людьми як результат набутих практичних знань і є приблизними в тому сенсі, що вони не потребують вичерпної попередньої інформації та забезпечують певну впевненість (або невизначеність), що запропоноване рішення є правильним.

ЕС відрізняється від інших програм штучного інтелекту насамперед тим, що:

- ЕС мають справу з об'єктами реального світу, для роботи з якими зазвичай потрібен великий людський досвід. Більшість програм штучного інтелекту мають суто дослідницький характер, вони зосереджуються на абстрактних математичних проблемах або спрощених версіях проблем реального світу з метою підвищення інтуїції або розробки методологій. ЕС мають чітко виражену практичну спрямованість у сфері науки чи комерції;
- однією з найважливіших характеристик ЕС є її продуктивність, тобто швидкість отримання результату та рівень його достовірності (надійності). Програми штучного інтелекту можуть бути повільними та робити помилкові припущення в певних ситуаціях, оскільки вони більше схожі на інструмент дослідження, ніж програмне забезпечення. ЕС має знайти рішення не гірше того, яке пропонує експерт у цій ситуації, у прийнятний термін;
- ЕС необхідно мати можливість пояснити, чому було прийнято таке рішення, і довести його обґрунтованість. Ви маєте отримати всю необхідну інформацію для того, щоб рішення було прийнято правильно та професійно. Навпаки, програми ШІ зазвичай спілкуються лише зі своїми творцями, які вже розуміють, на чому базується результат. ЕС призначена для взаємодії з різними користувачами, для яких його робота має бути максимально прозорою.

Тому, на відміну від звичайних інтелектуальних програм, можна сказати, що ЕС – це система, яка базується на знаннях, а не на алгоритмічних чи статистичних методах. Це програма, яка може проводити повноцінний експертний аналіз у цій галузі. Тому процес створення ЕС навіть називають інженерією знань, а не програмуванням.

Враховуючи все вищесказане, можна визначити основні конструктивні елементи класичної ЕС (рис. 5.1).

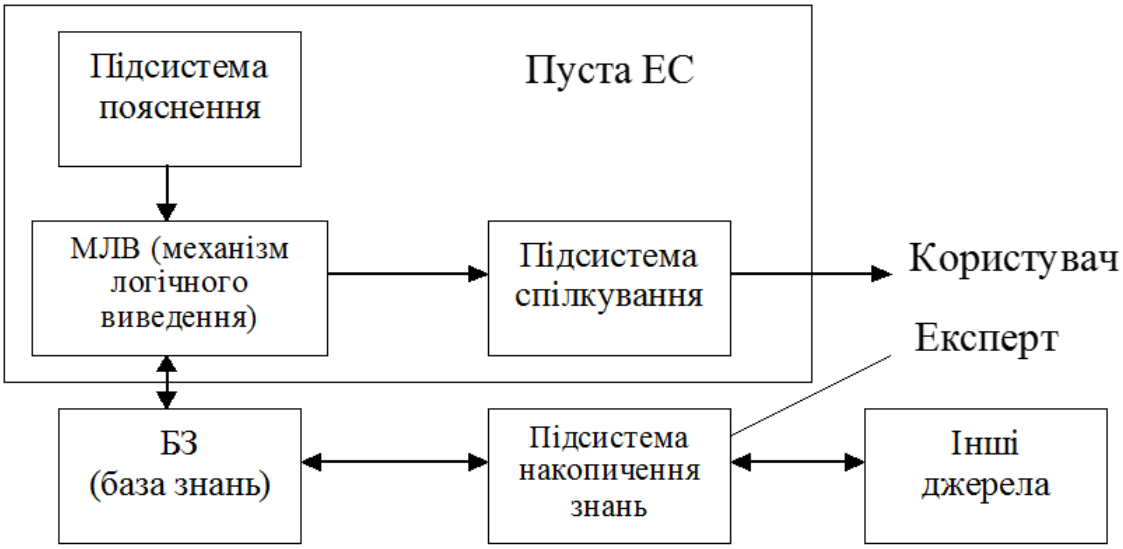


Рисунок 5.1 – Класична структура експертних систем

Визначимо основні терміни наведеної структури [22, 23].

Користувач є фахівцем у конкретному програмному забезпеченні, для якого призначена система.

Його кваліфікація недостатньо висока, тому йому потрібна допомога і підтримки з боку ЕС.

Експерта доцільно представити в подвійній ролі – ролі фахівця з ШІ (як проміжна ланка між базою знань і експертом - *підсистема накопичення знань*) і роль експерта.

Інтерфейс користувача – це *комунікаційна підсистема (підсистема спілкування)*: набір програм, які призначені для здійснення діалогу користувача з ЕС вже на етапі введення інформації, а також для отримання інформації.

Підсистема пояснень – це програма, яка дозволяє користувачеві отримати відповіді на питання «Як було отримано та чи іншу рекомендацію?» і «Чому система прийняла таке рішення?» Відповідь на питання «Як?» – відстеження всього процесу прийняття рішення із зазначенням використаних фрагментів БЗ. Відповідь на питання «Чому?» – посилання на заяву, яка безпосередньо передує отриманому рішенню.

Підсистема накопичення знань (інтелектуальний редактор) – це програма, яка надає інженеру знань можливість створювати базу знань у діалоговому режимі.

Механізм логічного виведення (МЛВ) – це програма, яка моделює міркування експерта на основі знань, що містяться в базі знань.

База знань (БЗ) – серце ЕС, сукупність знань у певній предметній області, збережені на машинному носії у зрозумілій фахівцеві та користувачеві формі. Паралельно з цим «людським» поданням існує БЗ в машинному поданні [22, 23].

5.2 Класифікація сучасних інформаційних технологій побудови експертних систем

Під час розробки експертних систем програмісти використовують спеціальні інструменти, які дозволяють суттєво скоротити час розробки. До таких інструментів відносяться як програмні, так і апаратні засоби.

До апаратних засобів зазвичай відносять: ПК, інтелектуальні робочі станції, портативні комп'ютерні системи та паралельні високопродуктивні комп'ютерні системи.

Загальну класифікацію програмного інструментарію можна показати так [22]:

1) процедурні мови, орієнтовані на обробку деякої символьної інформації (ЛІСП, АЛГОЛ тощо);

2) мови інженерії знань, іншими словами мови програмування високого рівня, що орієнтовані на розробку ЕС (PROLOG);

3) засоби автоматизованого конструювання, функціонування та модифікування ЕС (HEARSAY-4, RLL тощо);

4) «Оболонки» або базові (пусті) ЕС, що не містять баз знань (Decision Support Expert::Shell, Мала ЕС тощо).

У поданій класифікації інструментальні засоби (ІЗ) розташовані в порядку зменшення затрат праці, необхідних за їх використання для створення конкретних ЕС. У разі використання засобів першого типу програміст змушений самостійно програмувати всі компоненти ЕС мовою досить низького рівня. На другому рівні продуктивність різко зростає, але за рахунок деякого зниження продуктивності. ІЗ третього рівня дозволяють розробнику ЕС не розробляти всі або деякі компоненти ЕС, а вибирати їх із раніше створеного набору. Використовуючи ІЗ четвертого рівня, програміст повністю звільняється від роботи зі створення програм, оскільки в його розпорядженні є порожня система, яку необхідно заповнити знаннями про відповідне програмне забезпечення.

У той самий час за використання ІЗ 3-го і 4-го типів виникає ряд проблем:

- стратегії управління продуктивністю, які вони впроваджують, не можуть слідувати експертним методам вирішення, що може призвести до неефективних або зовсім неправильних рішень;

- мова подання знань може не підходити для цього програмного забезпечення.

Нині широкий вжиток отримали ІЗ, які мають назву «налагоджувальні оболонки». Вони дозволяють використовувати покриття не в якомусь «застиглому» вигляді, а генеруючи його на основі певний набір механізмів, передбачених в ЕС (ЕКО, НЕКС, ЕКСПЕРТ-ТИЗА). Останнім часом почали розрізняти оболонки, які називаються «оболонками додатків», і оболонками знань. Крім того, з'явилися універсальні та спеціалізовані оболонки.

Тому ми розглянемо інструментальні заходи проектування ЕС як системи програмування, що спрощують проектування ЕС. Їх склад і структура визначаються особливостями задач, які вирішують експертні системи та технологій проектування. Потрібно зазначити, що вибір технологій та інструментальних засобів реалізації ЕС – це ключове питання створення ЕС.

Відповідно до наведеної загальної класифікації розглянемо більш детально функції програмних засобів, що використовуються для розробки ЕС.

Традиційні мови програмування

До цієї групи засобів входять традиційні мови програмування (С, С++ тощо), які головно зосереджуються на числельних алгоритмах і які не дуже ефективні під час роботи з символами та логічними даними.

Побудова ЕС на основі цих мов потребує від програмістів кропітливої та віртуозної роботи. Але перевага цих мов – це потужна ефективність завдяки близькості традиційній машинній архітектурі. Крім того, використання традиційних мов програмування дозволяє вносити інтегровані ЕС у великі програмні комплекси. Найбільш зручними мовами в цьому плані є об'єктно-орієнтовані мови, особливо С++. Це пояснюється тим, що парадигма об'єктно-орієнтованого програмування тісно пов'язана зі структурою подання знань – фреймовою моделлю.

Мови штучного інтелекту

Одна з тенденцій еволюції систем програмування пов'язана з використанням мов функціонального програмування. Перша найпоширеніша мова такого типу була створена в 1959 році і називалася LISP (LISP – List Processing Language). Вона зосереджена на вирішенні проблем обробки символьної інформації.

Розвиток мов функціонального програмування відбувався шляхом реалізації в них механізмів роботи з БЗ. Прикладом такої мови є PROLOG (PROLOG – Programming Language Based on Logic), розроблена у 1971 р. і призначена для роботи з логічними БЗ. Універсальність цункціональних мов програмування невисока, але це компенсується різноманітними можливостями під час роботи з символьною та логічною інформаціями [22].

Спеціалізований програмний інструментарій

Ця група програмного забезпечення ШІ містить бібліотеки та додаткові компоненти мовами ШІ (LISP), які дозволяють користувачам працювати з оболонками ЕС на більш високому рівні, ніж це можливо за використання мовах штучного інтелекту.

«Оболонки»

«Оболонка» (Shells) – це «порожня версія» існуючих ЕС, це готові ЕС без розвиненої та заповленої бази знань. Перевагою оболонок є те, що для створення готової експертної системи вони взагалі не потребують навичок програміста. Для створення БЗ потрібні лише спеціалісти цієї галузі. Однак, якщо певна тематична область не підходить під модель, яка використовується в тій чи іншій оболонці, наповнити БЗ в цьому випадку досить складно або практично неможливо.

Деякі дослідники штучного інтелекту пропонують іншу класифікацію ІЗ. На їхню думку, інструментальні засоби можна умовно поділити на такі групи:

1. Способи організації даних і знань.
2. Засоби для підтримки.
3. Засоби сервісного обслуговування.
4. Інтегровані заходи.
5. Комплексні заходи.

Засобами для організації даних і знань є символічні мови програмування, непроцедурні мови програмування, мови інженерії знань (моделі подання знань), інформаційно-пошукові системи та системи керування базами даних (СКБД).

До з*асобів підтримки* відносяться асемблери і макроасемблери, інтерпретатори, компілятори, налагоджувачі і т. д.

Сервісні засоби – це редактори, бізнес-ігри та навчальні системи. Вони носять службово-допоміжний характер.

Комплексні засоби – СКБД з вбудованою інтерпретувальною та/або компілювальною мовою програмування, системи обробки даних із середовищем, спеціалізовані засоби обробки даних.

Інтегровані засоби – це інтегровані прикладні системи, «пусті» або інструментовані ЕС (ЕС-оболонки), середовища підтримки.

Це найскладніший і найефективніший інструмент для реалізації практично всіх додатків ЕС.

Основними компонентами цих систем є обробка тексту і таблиць, засоби управління базами даних і передові графічні засоби. Такі системи надають користувачеві власне операційне середовище, яке дозволяє вирішувати проблеми без безпосереднього використання операційної системи.

Для створення прикладних ЕС ефективно використовуються оболонки ЕС, що підтримують відомі моделі подання знань – продукти, фрейми, семантичні мережі, дерева, таблиці тощо.

Нарешті, найпотужнішим засобом є середовище підтримки, що поєднує засоби автоматизації для всіх основних етапів створення ЕС. Більшість середовищ створено на основі нової концепції життєвого циклу програмного забезпечення. Під життєвим циклом розуміється послідовність технологічних етапів створення, експлуатації та розробки програмного забезпечення [22].

5.3 Об'єктно-орієнтоване програмування під час конструювання експертних систем

Питання про те, як допомогти користувачу зрозуміти структуру та функції деякого складного компонента програми, пов'язане з порівняно новою областю взаємодії людини і машини, що виникла на перетині таких сфер, як штучний інтелект, промислова технологія, фізіологія та ергономіка. Дослідники, які займаються експертними системами, розробляють методи подання інформації про поводження програми в процесі формування ланцюжка логічних висновків під час пошуку рішення [22, 23].

Подання інформації про поведінку експертної системи важливо з багатьох причин:

- *користувачі*, які працюють із системою, мають потребу в підтвердженні того, що у кожному конкретному випадку висновок, до якого дійшла програма, переважно коректний;
- *інженери, які формують базу знань*, мають переконатися, що сформульовані ними знання, застосовані правильно, зокрема й у випадку, коли існує прототип;
- *експертам у предметній області* бажано простежити хід міркувань і спосіб використання тих зведень, що з їх слів були введені в базу знань. Це дасть змогу судити, наскільки коректно вони застосовуються в цій ситуації;
- *програмістам,* які супроводжують, налагоджують і модернізують систему, потрібно мати у своєму розпорядженні певний інструмент;
- *менеджери системи,* які використовують експертну технологію, що зрештою несуть відповідальність за наслідки рішення, прийнятого програмою, також мають потребу в підтвердженні, що ці рішення достатньо обґрунтовані.

Об'єктно-орієнтований підхід до проектування – це метод, що логічно приводить до об'єктно-орієнтованої декомпозиції.

Під час застосування об'єктно-орієнтованого проектування створюються гнучкі програми, написані ощадливими засобами. За розумного поділу простору станів досягають більшої впевненості в правильності програми. У підсумку, зменшується ризик під час розроблення складних програмних систем.

Побудова моделей надто важлива у проектуванні складних систем.

Об’єктно-орієнтоване проектування пропонує багатий вибір моделей:

- динамічна модель;
- статична модель;
- логічна модель (структура класів, структура об’єктів);
- фізична модель (архітектура модулів, архітектура процесів).

Об’єктно-орієнтоване проектування відбиває ієрархію і класів, і об’єктів системи. Ці моделі охоплюють увесь спектр найважливіших конструкторських рішень, які необхідно розглядати під час розроблення експертної системи, і в такий спосіб спонукають створювати проекти, що мають усі атрибути добре організованих складних систем.

Основна складність у використанні засобів об’єктно-орієнтованого програмування – з’ясувати для себе, що саме має становити програмний об’єкт відносно предметної області. У ранніх версіях об’єктно-орієнтованих мов, які були призначені переважно для розробки програм моделювання, така проблема не виникала, тому що програмні об’єкти були об’єктами модельованої системи. Наприклад, під час моделювання виробничої лінії окремі програмні об’єкти являли собою ті або інші механізми цієї лінії, а повідомлення між програмними об’єктами – інформаційні, енергетичні і матеріальні потоки. Завдання програміста серйозно полегшувалося тим, що існувала достатньо очевидна відповідність між програмними і реальними об’єктами.

Але для того, щоб упровадити об’єктно-орієнтований стиль в проектування експертних систем, потрібно задуматися над тим, як співвіднести програмні об’єкти з абстрактними поняттями і категоріями предметної області. Об’єкти мають являти собою факти і цілі, набори правил або окремі гіпотези. Тому далеко не очевидно, якими повідомленнями мають обмінюватися такі об’єкти і який сенс має вкладатися в ці повідомлення.

Багато, що залежить від того, на якому рівні абстракції використовуватиметься об’єктно-орієнтований механізм. Якщо об’єкти є низькорівневою реалізацією певної схеми формування думок, то відпадає необхідність у використанні будь-яких епістемологічних послідовностей. Якщо ж об’єкти будуть видимі і для експерта в процесі розробки та вдосконалення системи, і для користувача під час експлуатації системи, то схема відображення понять і категорій на програмні об’єкти має бути ретельно продумана.

5.4 Технології експертних систем у СППР

Найбільший прогрес серед комп'ютерних інформаційних систем відзначено в сфері розробки експертних систем, основаних на використанні штучного інтелекту. Експертні системи дають можливість менеджеру або спеціалісту отримувати консультації експертів з будь-яких проблем, про які цими системами накопичені знання [20].

Під штучним інтелектом зазвичай розуміють здатності комп'ютерних

систем до таких дій, які називалися б інтелектуальними, якби виходили від людини. Рішення спеціальних завдань потребує спеціальних знань. Однак не кожна компанія може собі дозволити тримати у своєму штаті експертів з усіх, пов'язаних з її роботою, проблем або навіть запрошувати їх щоразу, коли проблема виникла.

Головна ідея використання технології експертних систем полягає в тому, щоб отримати від експерта його знання і, завантаживши їх у пам'ять комп'ютера, використовувати всякий раз, коли в цьому виникне необхідність. Будучи одним з основних додатків штучного інтелекту, експертні системи являють собою комп'ютерні програми, що трансформують досвід експертів в якій-небудь галузі знань у форму евристичних правил (евристик).

Евристики не гарантують отримання оптимального результату з такою ж упевненістю, як звичайні алгоритми, використовувані для вирішення завдань в рамках технології підтримки прийняття рішень. Однак часто вони дають достатньою мірою прийнятні рішення для їх практичного використання. Все це робить можливим використовувати технологію експертних систем як систем, що дають поради.

Крім того, експертні системи нині мають багато обмежень і недоліків. З погляду підтримки створення рішень головний їх недолік полягає в тому, що вони *не забезпечують підтримку обраних рішень, оскільки сама експертна система створює рішення, відтворюючи аналітичну логіку людини-експерта.* ОПР, зі свого боку, може прийняти або не прийняти ці рішення залежно від наявної ситуації і діючих факторів, не врахованих експертною системою (ЕС). На відміну від цього, СППР допомагає ОПР створювати рішення. Звідси випливає, що ЕС не є додатковою системою забезпечення підтримки прийняття рішень.

Перерахуємо деякі обмеження, що є недоліками ЕС:

- ЕС зазвичай працюють тільки у вузько визначених проблемних доменах, їхній рівень розуміння середовища, в якому вони функціонують, є певною мірою поверхневим;
- ці системи донині так і не мають здатності «здорового глузду», як інструментальні засоби вони, зазвичай, не здатні обмірковувати проблему багатьма способами або на різних рівнях;
- ЕС не можуть самі навчатися;
- успішні ЕС можуть привести до реальних змінень методики виконання своїх завдань людиною. Це може потребувати значних організаційних і технологічних змін, які можуть стати на заваді повному успіху системи, навіть якщо вона технічно досконала.

Незважаючи на ці обмеження, багато корпорацій розробили як експериментальні, так і діючі програми ЕС.

Питання для самоконтролю

1. Наведіть та охарактеризуйте класичну структуру ЕС.

2. Охарактеризуйте взаємозв’язок експертних систем (ЕС) та СШІ.

3. Поясніть значення терміна «експертний аналіз» та наведіть приклади його практичного застосування.

4. Наведіть основні характеристики ЕС, які відрізняють їх від звичайних програм.

5. Наведіть основні характеристики ЕС, які відрізняють їх від програм штучного інтелекту.

6. Наведіть класифікацію інструментальних засобів побудови ЕС та охарактеризуйте її.

7. Розкрийте суть об’єктно-орієнтованого програмування (ООП) під час конструювання ЕС.

8. Які протиріччя існують під час конструювання ЕС внаслідок застосування ООП?

9. Яка вагомість застосування технології ЕС у СППР.

10. Наведіть недоліки та обмеження в застосуванні ЕС.

6 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

6.1 Загальна характеристика технологій інтелектуального аналізу даних

Концепція *інтелектуального аналізу даних* (ІАД) об’єднує групу технологій, метою яких є отримання знань із даних, тобто виявлення у вихідних даних раніше невідомих нетривіальних, практичних та інтерпретованих знань, необхідних для прийняття рішень у багатьох предметних галузях [21].

Вибір методу ІАД насамперед визначається типом знань, які ми хочемо отримати, і способом подання вихідних даних. Якщо тип знань, отриманий за допомогою технології ІАД, є закономірностями предметної області, важливо, який тип спостережуваних подій домінує під час вибору методу ІАД (будуть домінувати випадкові чи причинно-наслідкові події; чи обидва типи подій спостерігаються одночасно).

Методи ІАД оперують даними, поданими: а) атрибутивно (об'єкти описуються значеннями певного фіксованого набору атрибутів); б) структурно (об'єкти визначаються за типологією); в) повнотекстово (вихідними даними є тексти природною мовою).

Існує три основних класи методів ІАД: а) алгебра (вихідні дані мають подаватися у вигляді структур алгебри), б) статистичні (для цього використовують апарат теорії ймовірностей і математичної статистики); в) методи м'якого обчислення (використовується нечітке подання даних і нейронні мережі).

Можна вважати, що вихідні дані в ІАД подано в цифровій або символьній формі. В цьому випадку до 80 % усіх даних існує в неструктурованому вигляді (містяться в текстових документах), що визначає важливість інтеграції засобів ІАД та аналізу тексту в технології управління знаннями. Методи ІАД реалізовано в трьох основних технологіях: технологіях інтерактивної аналітичної обробки даних (On-Line Analytical Processing − OLAP); технологіях глибокого аналізу або отримання даних (Data Mining − DM) і технології візуалізації даних.

6.2 Технологія інтерактивної аналітичної обробки даних

Технологія інтерактивної аналітичної обробки даних (OLAP) переважно зосереджена на обробці невиконаних запитів до сховищ даних. Поява сховищ даних багато в чому пов'язана з тим, що прямий аналіз систем OLTP (On-Line Transactional Processing) неможливий або ускладнений через неоднорідність, різні формати та розподіл вихідних даних у корпорати-

вній мережі. Репозиторії даних гарантують, що вони зберігаються в одному місці в межах доступної структури реалізації програми. Принципи OLAP для якісного та кількісного оцінення результатів і динаміки діяльності підприємства, інформація про яке розміщується в сховищі даних, сформулював Е. Код. Основним принципом є підтримка багатовимірного подання даних. У багатовимірній моделі даних база даних має форму одного або кількох кубів даних (гіперкубів), осі яких становлять основні атрибути аналізованого бізнес-процесу. На перетині осей, у комірці гіперкуба, знаходяться дані (міри, показники), які кількісно характеризують аналізований процес [21].

У процесі аналізування виконуються певні операції побудови перетинів гіперкуба шляхом визначення значень атрибутивно-координатних множин; операції стиснення гіперкуба внаслідок переходу до значень атрибутів-вимірів вищих рівнів ієрархії та відповідного агрегування значень пов'язаних з ними показників; операції щодо деталізації даних обернені до операцій стиснення. Для деяких атрибутів закономірно виведено ієрархічні зв'язки: наприклад, для атрибута «час» ієрархія така: роки – квартали – місяці, для атрибута «територія»: області – міста – округи тощо. Зручність сприйняття даних для аналітика забезпечується обертанням куба зі зміною порядку вимірювань. Візуалізація даних гіперкуба переважно здійснюється за допомогою двовимірних подань у формі таблиць зі складними ієрархічними заголовками рядків і стовпців.

Багатовимірність у додатках OLAP реалізована в рамках дво- або трирівневої архітектури. Перший рівень містить засоби для візуалізації багатовимірних даних і маніпуляції для кінцевого користувача та підтримує багатовимірне подання даних, абстрагованих від їх фізичної структури. Інструменти першого рівня містять, наприклад, клієнти зведеної таблиці OLAP від Microsoft Excel 2007 і сервери OLAP, зокрема Oracle Express Server і Microsoft OLAP Services. Другий рівень містить спеціальну мову для формулювання багатовимірних запитів, відмінну від SQL, і програмний процесор здатний їх виконувати, забезпечуючи обробку багатовимірних даних. Цей рівень найчастіше вбудований в OLAP-клієнт або OLAP-сервер, хоча може існувати і як самостійний продукт (наприклад, служба Microsoft Pivot Tables). На третьому рівні реалізується фізична організація багатовимірного сховища даних за допомогою звичайної реляційної або спеціальної багатовимірної OLAP-СУБД. Додатки OLAP найчастіше використовують комбінацію цих засобів, зокрема: MOLAP (Multidimensional OLAP), а окремі факти та агрегати даних зберігаються в багатовимірній базі даних; у ROLAP (Relational OLAP) окремі факти зберігаються в реляційній базі даних, а агрегати – у спеціально створених службових таблицях; у HOLAP (Hybrid OLAP) окремі факти зберігаються в реляційній базі даних, агрегати зберігаються в багатовимірній базі даних.

6.3 Технологія добування даних

Поняття *дейтамайнінгу* (Data Mining − DM) чи глибинного аналізу даних, об'єднує інструментальні програмні засоби добування корисної інформації з накопичених у електронному вигляді обсягів «сирих» даних за допомогою виявлення прихованих від користувача шаблонів чи зразків (patterns) зв'язків між даними [21].

На відміну від аналізу даних суто статистичними методами, який відбувається в «режимі перевірки» (verification mode) (спочатку формулюється гіпотеза про певний зв’язок між даними, яка підтверджується або відхиляється аналізом даних, отриманих на основі запиту до бази даних), програма, що забезпечує DM, працює в «режимі відкриття» (discovery mode) без перевірки попередньо сформованої гіпотези про зв'язки між даними, і виявляє приховані регулярні зв'язки між ними, які називаються шаблонами (patterns). Нині дейтамайнінг подано сімейством комерційних програмних продуктів (аналітичних додатків), які підтримують прийняття рішень на основі пошуку прихованих шаблонів у базах даних або сховищах даних з доставкою важливих фрагментів інформації або їх результатів у відповідь на аналіз інформаційних запитів користувачів у зручній формі (схеми, діаграми, зведені таблиці, звіти тощо). Водночас у технології DM під час автоматичного аналізу даних ініціатива генерації зразків (шаблонів) належить інтелектуальній системі.

Англійський термін Data Mining перекладається як «видобуток» або «вилучення даних» і часто пояснюється як пошук, аналіз, інтерпретація та подання інформації зі сховищ, баз даних або Інтернет. Інтелектуальний аналіз даних переважно передбачає фільтрацію великих обсягів даних для вибору інформації, необхідної для прийняття конкретного рішення. Корпорація IBM трактує дейтамайнінг як «процес вилучення раніше невідомої, але важливої інформації з великих баз даних, що дозволяє її обробити та використовувати для прийняття ключових рішень у бізнесі», у вітчизняній літературі як аналог цього терміна іноді вживається термін «інтелектуальні обчислення». Якщо дослідження даних відбувається в Інтернеті, також використовується термін Data Surfing.

Основним призначенням технології DM є аналіз структурованих даних за допомогою математичних моделей, основаних на статистичних, імовірнісних і оптимізаційних методах, з метою виявлення у них заздалегідь не відомих закономірностей, залежностей і добування непередбаченої інформації, тобто для отримання нових корисних знань із загальних масивів інформації.

Основні завдання, які підтримує DM, містять класифікацію, групування, пошук асоціацій (завдання встановлення залежностей) і кореляцій, виявлення типових шаблонів у заданому наборі, виявлення аномалій (об’єктів даних, які не відповідають встановленим характеристикам і пове-

дінці), дослідження тенденцій у часових рядах з побудовою відповідних регресійних моделей для прогнозування майбутніх ситуацій, регресії тощо. *Класифікація* передбачає виявлення ознак, що характеризують групу, до якої належить цей об’єкт, шляхом аналізу вже класифікованих об’єктів і формулювання певного набору правил та/або ідентифікація групи, до якої належить цей об'єкт, поширена на цю властивість групи, важливу для прийняття рішень. Наприклад, визначення категорії позичальника за ризиком повернення кредиту гарантує обґрунтоване рішення про надання йому кредиту взагалі та на умовах кредиту. *Кластеризація* передбачає розподіл об'єктів на раніше невідомі групи на основі подібності або близькості значень певних ознак. *Асоціації* та *кореляції* стосуються групування елементів даних на основі виявлення взаємозв’язків між ними (наприклад, визначення того, що дії одного типу, як показує аналіз масиву даних, здебільшого супроводжуються діями іншого типу або навпаки). Виявлення типових зразків ґрунтується на заданих продукційних правилах «якщо А, то Б», виявлених аналізом раніше встановлених закономірностей. *Задача регресії* багато в чому схожа з завданням класифікації, але в ході її рішення проводиться пошук шаблонів для визначення числового значення. Іншими словами, параметр, що передбачається тут, як правило, число з безперервного діапазону [21].

Окремо виділяється задача прогнозування нових значень на підставі наявних значень числової послідовності (або декількох послідовностей, між значеннями в яких спостерігається кореляція). Водночас можуть враховуватися наявні тенденції (тренди), сезонність, інші чинники. Класичним прикладом є прогнозування цін акцій на біржі.

У табл. 6.1 наведено приклади задач інтелектуального аналізу даних з різних сфер.

За способом вирішення задачі інтелектуального аналізу можна розділити на два класи: навчання з учителем (від англ. supervised learning) і навчання без вчителя (від англ. unsupervised learning). У першому випадку потрібен навчальний набір даних, на якому створюється і навчається модель інтелектуального аналізу даних. Готова модель тестується і згодом використовується для передбачення значень в нових наборах даних. Іноді в цьому ж випадку говорять про керовані алгоритми інтелектуального аналізу. Задача класифікації і регресії відносяться саме до цього типу.

У другому випадку метою є виявлення закономірностей наявних в існуючому наборі даних. В цьому випадку навчальна вибірка не потрібна. Як приклад можна навести задачу аналізу споживчого кошика, коли в процесі дослідження виявляються товари, які найчастіше купуються разом. До цього ж класу належить задача кластеризації.

Таблиця 6.1 − Приклади застосування інтелектуального аналізу даних

	Інформаційні технології	**Торгівля**	**Фінансова сфера**
Класифікація			Оцінка кредито-спроможності
Регресія			Оцінка допустимого кредитного ліміту.
Прогнозування		Прогнозування про-дажів	Прогнозування цін акцій
Кластеризація		Сегментація клієнтів	Сегментація клієнтів
Визначення взаємозв'язків		Аналіз споживчого кошика	
Аналіз послідовностей	Аналіз переходів по сторінках web-сайту		
Аналіз відхилень	Виявлення вторгнень в інформаційні системи		Виявлення шахрайства з банків-ськими картками

Також можна говорити про класифікацію задач інтелектуального аналізу даних за призначенням, відповідно до якої вони діляться на описові (descriptive) і передбачувальні (predictive). Мета рішення описових задач − краще зрозуміти досліджувані дані, виявити наявні в них закономірності, навіть якщо в інших наборах даних вони зустрічатися не будуть. Для передбачувальних задач характерно те, що під час їх вирішення на підставі набору даних з відомими результатами будується модель для передбачення нових значень.

Загалом, відповідно до використовуваних моделей обробки даних та отриманих результатів, процеси інтелектуального аналізу даних поділяються на три групи: відкриття або добування *(discovery),* моделювання прогнозів *(predictive modeling)* і аналіз аномалій *(forensic analysis)*. Процеси виявлення передбачають просіювання ряду інформації для виявлення невідомих прихованих шаблонів без будь-якої попередньої гіпотези щодо їх наявності чи природи на основі запитів, створених системою на основі критерію відповідності інтересів користувача проблемі, що вирішується. Під час моделювання прогнозів (моделювання наслідків або прогнозування трендів) з інформаційної таблиці беруться і обробляються зразки, які система вважає необхідними для отримання прогнозу, тобто нових значень даних в нових умовах. Аналіз аномалій – це процес використання обраних шаблонів для виявлення аномалій (незвичайних, нетипових елементів даних) на основі визначення системою норми та допустимого рівня відхилень від неї.

Щоб розв'язати всі ці задачі, потрібно обробити великі обсяги інформації. Створення алгоритмів для їх вирішення необхідно враховувати ор-

ганізацію джерел даних, їх значний обсяг, великі розміри завдань і забезпечити масштабованість алгоритмів. У DM штучні нейронні мережі та методи кластерного аналізу використовуються для *сегментації даних*, дерева рішень, генетичні алгоритми та методи нечіткої логіки використовуються для *індуктивного виведення*, методи нечіткої логіки та генетичні алгоритми – для *виявлення* в інформаційному масиві певних пар об'єктів; статистичні методи (кореляційний та регресійний аналіз) та асоціативні методи (метод «найближчого сусіда»), методи припущення асоціацій тощо) – для об’єктів, які часто трапляються. Для подання отриманих результатів (дистильованих даних) використовуються візуалізація та крос-табуляція (подання даних у крос-таблицях).

Кластерний аналіз (таксономія) – це спосіб групування багатовимірних даних, значення котрих подаються точками багатовимірного геометричного простору, в однорідні підмножини (групи, «грона», скупчення, кластери) так, щоб точки всередині таких груп були схожими за певними ознаками у багатовимірному просторі ознак («близькими»), а точки з різних груп – відповідно несхожими. Таким чином за застосування методу *дерев рішень* для навчальної тестової вибірки даних створюється ієрархічна структура правил класифікації такого типу: «ЯКЩО... ТОДІ...», що і має вигляд дерева. Для того щоб вирішити, до якого класу зарахувати той чи інший об'єкт або ситуацію, потрібно відповісти на запитання, які стоять у вузлах такого дерева, починаючи з його кореня. Після обробки останньої гілки визначається тип об'єкта або ситуації та витягуються з бази даних рекомендовані правила їх обробки для отримання найбільш корисного або найменш шкідливого ефекту. Концепція *генетичних алгоритмів* запозичена в живій природі та полягає в комп’ютерному моделюванні еволюційного процесу створення, модифікації, відбору та оптимізації найкращих рішень, які в процесі подальшого розвитку та модифікації (відбору) здатні генерувати ще кращі рішення (кращих «нащадків») так, як це відбувається в механізмах генетичної спадковості і природного відбору. Ідея методу *«найближчого сусіда»* (тобто методу міркування, основаного на подібних випадках) полягає в тому, щоб знайти близьку аналогію поточної ситуації (подібний набір даних) у минулому (в попередньо збережених рядах даних) і виберіть той самий наслідок, який спостерігався для аналога, тобто правильну відповідь для аналога. Алгоритми виявлення асоціацій здійснюють пошук правил одночасної появи окремих об'єктів, їх властивостей або способів їх прояву (поведінки) в конкретних умовах. Схема процесу ІАД за технологією DM складається з чотирьох основних етапів (рис. 6.1). На першому етапі проблема формулюється на основі цільових змінних. На другому етапі дані для аналізу готуються у вигляді таблиці, рядки (записи) якої відповідають об’єктам або їх станам, а стовпці (поля, змінні) – властивостям (характеристикам) об’єктів. З набору властивостей видаляються зайві та безінформаційні елементи, тобто такі, що мають однакові значення

майже для всіх записів, а також властивості, кількість значень яких схожа на записи. Також видаляються записи рідкісних особливих ситуацій (якщо їх виявлення не є метою аналізу) і неправильні або дуже неточні записи значень, які можуть істотно негативно вплинути на результати аналізу. На третьому етапі здійснюється аналіз власне даних методами DM. Змістом четвертого етапу є перевірка та інтерпретація отриманих результатів (витягнутих знань). Для верифікації застосовують тестовий набір записів, які виділено з вихідних даних, але не проаналізовано [21].

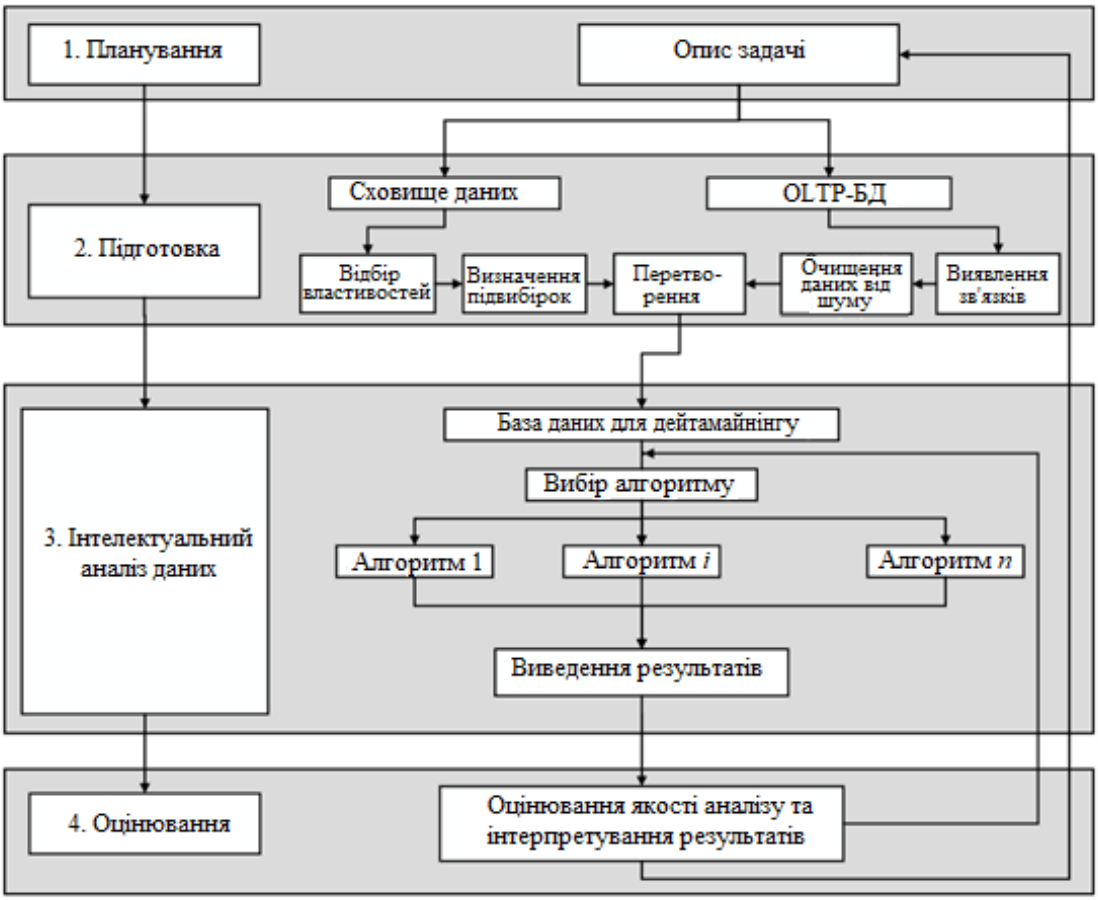


Рисунок 6.1 − Інтелектуальний аналіз даних за технологією DM

На цей момент програмне забезпечення дейтамайнінгу подано значною кількістю програмних продуктів, серед яких: Intelligent Miner від IBM на основі ШНМ, прогнозне моделювання, виявлення асоціацій тощо; Decision Series від Neo Vista Software на основі ШНМ, правила асоціації, дерева рішень і кластерів; система Knowledge Studio сімейства програмного забезпечення Angoss на основі ШНМ, різні алгоритми дерева рішень і кластерного аналізу; система MineSet від Silicon Graphics, основана на численних алгоритмах інтелектуального аналізу даних і унікальній технології для візуалізації зв’язків у багатовимірних базах даних. Практично всі ці системи дозволяють виявляти багатофакторні залежності у вигляді функціональних

виразів, створювати структурні та класифікаційні правила, прогнозувати та візуалізувати виявлені закономірності.

6.3.1 Методи і моделі Data Mining

Традиційно мали місце два типи статистичних аналізів: *підтверджувальний аналіз (confirmatory analysis)* та *дослідницький аналіз (exploratory analysis).*

У підтверджувальному аналізі кожний користувач має конкретну гіпотезу і внаслідок аналізу або підтверджує, або спростовує її. Однак недоліком підтверджувального аналізу є недостатня кількість гіпотез у аналітика. За дослідницького аналізу виявляються та підтверджуються або спростовуються схожі гіпотези. Тут уже сама система, а не її користувач, відповідає за аналіз даних [20].

Як правило, термін «дейтамайнінг» використовується для опису автоматизованого процесу аналізу даних, в якому система сама бере ініціативу щодо генерування взірців, тобто дейтамайнінг належить до інструментальних засобів, які мають дослідницький аналіз. З погляду орієнтації на процес є три типи таких процесів дейтамайнінгу (рис. 6.2): відкриття (добування) (discovery); моделювання прогнозувань (predictive modeling); аналіз аномалій (forensic analysis).

Відкриття **–** це процес сканування бази даних для пошуку невидимих шаблонів (pattern) без будь-яких упереджених ідей чи гіпотез щодо того, якими вони можуть бути. Іншими словами, програма бере на себе ініціативу без попереднього розгляду того, чи шаблони (зразки), що цікавлять користувачів, насправді існують і чи можуть бути подані у вигляді відповідних запитів.

У великих базах даних стільки інформаційних аспектів, що користувач взагалі майже не замислюється і не ставить правильних запитів щодо відповідних зразків. Ключовим питанням тут може бути кількість закономірностей, які можна виразити та виявити, а також якість отриманої інформації. Саме це і визначає потужність засобів відкриття інформації (discovery).

У ***моделюванні передбачень*** шаблони витягуються з бази даних, яку можна використовувати для прогнозування майбутнього.

Моделювання передбачень дає змогу користувачеві створювати записи з деякими невідомими дослідницькими значеннями, і система визначає ці невідомі значення, які ґрунтуються на попередніх шаблонах, що відкриваються з бази даних. У той час як відкриття знаходить шаблони в даних, прогнозне моделювання використовує шаблони для пошуку значень для нових елементів даних, і це головна відмінність між цими типами процесів дейтамайнінгу.

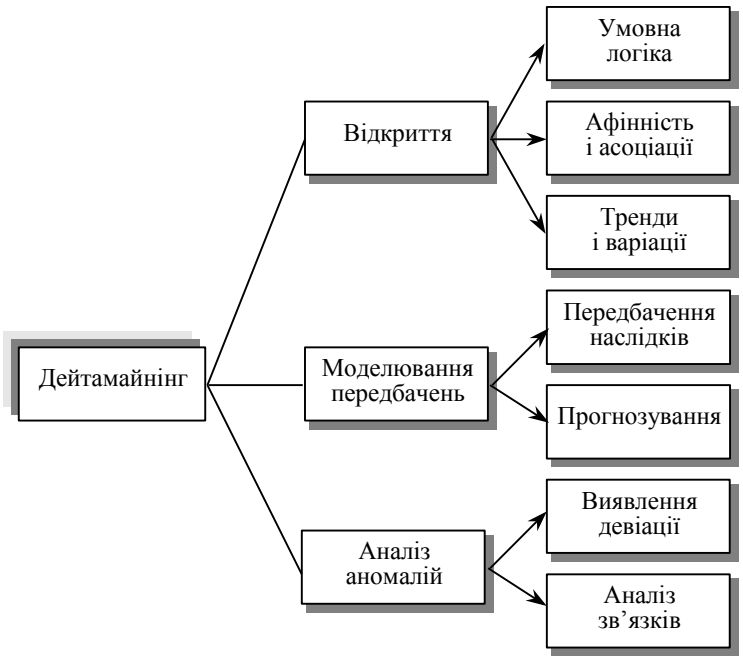


Рисунок 6.2 − Типи процесів дейтамайнінгу

Аналіз аномалій (forensic analysis) **–** це процес використання відібраних зразків (шаблонів) для виявлення аномалій чи незвичайних елементів даних. Щоб відрізнити нетипові елементи, необхідно спочатку з'ясувати, що є нормою, і тільки потім використовувати задані порогові значення для виявлення тих елементів, які відхиляються від звичайних. Зокрема, це стосується виявлення девітації, відхилень від правильного курсу [20].

Кожен із цих процесів можна додатково охарактеризувати, виділяючи відповідні методи. Наприклад, існує кілька методів виявлення закономірностей: правило *«ЯКЩО..., ТО», асоціації, спорідненість (безперервність)* і т. д. Якщо правило «ЯКЩО..., ТО» людині відомо, то правила асоціації є новинкою. Вони стосуються групування елементів даних (наприклад, коли хтось купує один продукт, за звичкою чи випадковістю він або вона може купити інший продукт одночасно; цей процес зазвичай містить аналіз ринкового кошика покупця). Потужність системи відкриття вимірюється кількістю типів і загальністю шаблонів, які можна знайти та виразити звичною для використання мовою.

Користувачі та діяльність з аналізу даних

Описані процеси потрібно відрізняти від діяльності дейтамайнінгу, за допомогою яких можуть бути реалізовані процеси інтелектуального аналізу даних, і користувачів, які виконують ці дії. Дії дейтамайнінгу, як правило, виконуються трьома різними типами користувачів: розробниками (executives), кінцевими користувачами (end users) та аналітиками (analysts). Усі користувачі зазвичай виконують три типи діяльності дейтамайнінгу в корпоративному середовищі: епізодичні; стратегічні та безперервні (пос-

тійні). Безперервна та стратегічна діяльність дейтамайнінгу часто безпосередньо залучає інших виконавців та менеджерів, хоча аналітики також можуть допомогти їм у цьому [20].

Мета технології Data Mining − знаходження в даних таких закономірностей, які не можуть бути знайдені традиційними методами. Є два види моделей: предиктивні та описові.

Предиктивні моделі будуються на основі набору даних з відомими результатами. Вони використовуються для прогнозу результатів на основі інших наборів даних. Необхідно, щоб модель працювала максимально точно, була статистично значимою і виправданою. До них належать *моделі класифікації* − описують правила або набір правил, відповідно до яких можна віднести опис будь-якого нового об'єкта до одного з класів. Такі правила будуються на основі інформації про наявні об'єкти шляхом поділу їх на класи; *моделі послідовностей* − описують функції, що дають змогу прогнозувати зміну параметрів. Вони будуються на основі даних про зміну певного параметра за минулий період часу.

Описові (descriptive) моделі пов'язані із залежностями в наборі даних, взаємним впливом різних чинників, тобто із побудовою емпіричних моделей різних систем. Ключовий момент у таких моделях − легкість і прозорість для сприйняття людиною. Можливо, знайдені закономірності будуть специфічною межею саме конкретних досліджуваних даних і більше ніде не зустрінуться, але все це може бути корисним. До них належать такі види моделей:

- кластеризації − описують групи (кластери), на які можна поділити об'єкти, дані про які піддаються аналізу. Групуються об'єкти (спостереження, події) на основі даних (властивостей), що описують суть об'єктів. Об'єкти усередині кластера мають бути подібними один до одного і відрізнятися від об'єктів, що ввійшли до складу інших кластерів;

- виключення − описують виняткові ситуації в записах, які різко відрізняються від основної множини записів;

- підсумкові (результатні) − виявлення обмежень на даних масиву аналізу. Подібні обмеження важливі для розуміння даних масиву, тобто це нове знання, здобуте внаслідок аналізу. Таким чином, Data Summarization − це знаходження будь-яких фактів, істинних для всіх або майже всіх записів у вибірці даних, що вивчається, але які досить рідко зустрічалися в усьому різноманітті записів такого самого формату;

- асоціації − виявлення закономірностей між пов'язаними подіями.

Для побудови розглянутих моделей використовуються різні методи та алгоритми Data Mining.

Технології дейтамайнінгу використовують велику кількість методів, частина з яких запозичена з інструментарію ШІ, а інша може належати до класичних статистичних методів чи до інноваційних методів, породжених останніми досягненнями інформаційних технологій. Вищий рівень класи-

фікації методів дейтамайнінгу може ґрунтуватися на тому, чи зберігаються дані після дейтамайнінгу незмінними, чи вони фільтруються для подальшого використання.

На рис. 6.3 показано дерево методів дейтамайнінгу, де відображені основні види і підвиди методів, причому гілкування можна продовжити, оскільки низка методів, наприклад, кластерний аналіз, нейромережі, дерева рішень мають багато різновидів. Оскільки деякі з наведених вище методів були розглянуті досить поверхово у світлі ідентифікації інструментарію штучного інтелекту або як структурні частини певних продуктів дейтамайнінгу, то зупинимося на короткому аналізі складових дерева методів дейтамайнінгу [20].

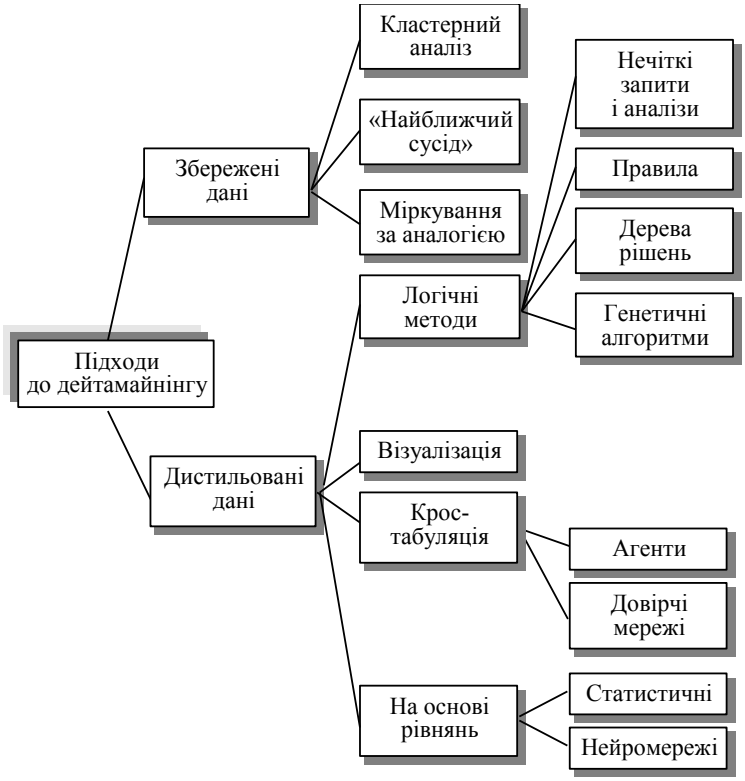


Рисунок 6.3 − Дерево методів дейтамайнінгу

Збереження даних (Data Retention)

Під час дистиляції шаблонів ми аналізуємо дані, виділяємо необхідний шаблон, а потім залишаємо його, використовуючи метод зберігання, дані зберігаються для порівняння з шаблоном. Коли надходять нові елементи даних, вони порівнюються з попереднім рядом даних.

Метод «найближчого сусіда» (схожий сусід або «nearest neighbor») є добре відомим прикладом підходу на основі зберігання. У той самий час ряд даних зберігається в пам'яті для порівняння з новими елементами даних. Коли подано новий запис для прогнозування, виявляються «відхилення» між ним і подібними рядками даних і ідентифікується найбільш схожий.

Міркування за аналогією (case-based reasoning − CBR) або *міркування на основі прецедентів (аналогічні випадки)*. Цей метод має дуже просту ідею: щоб зробити прогноз на майбутнє або прийняти правильне рішення, система CBR знаходить близькі аналогії в минулому за різних умов і вибирає правильну відповідь на основі подібних характеристик. Інструменти логічного висновку на основі випадків шукають у базі даних записи, схожі на описані записи. Користувач описує, наскільки міцним має бути зв'язок, перш ніж запропонувати нову справу. Ці типи інструментів також називають міркуванням на основі пам'яті (memory-based reasoning).

Кластерний аналіз – це метод групування багатовимірних об'єктів шляхом подання результатів окремих спостережень через точки в геометричному просторі з подальшим вибором груп як «кластерів» цих точок. Термін «кластерний аналіз» був запропонований К. Трайоном у 1939 році (cluster, *англ.* − гроно, скупчення, пучок). Вирази: *автоматична класифікація, таксономія, розпізнавання без навчання, розпізнавання образів без вчителя, самонавчання* тощо є синонімами (хоча і відносними). У дейтамайнінгу кластеризація використовується для класифікації (таксономії).

Основною метою кластерного аналізу є вибір однорідних підмножин із вихідних багатовимірних даних, щоб об'єкти всередині груп були схожі за певними характеристиками, а об'єкти з різних груп відрізнялися. «Схожі» означає близькість об'єктів у багатовимірному просторі ознак, і тоді завдання зводиться до вибору природних кластерів об'єктів у цьому просторі, які вважаються однорідними групами. У кластерному аналізі використовуються десятки різних алгоритмів і методів (один із них – метод K-Means, реалізований в системі дейтамайнінгу Knowledge STUDIO) [20].

Дистиляція даних (Data Distilled)

За допомогою цього методу шаблон (зразок) вибирається з низки даних і потім використовується для різних цілей. Тут природно виникають два запитання: 1. «З яких типів шаблонів ви можете вибрати?» і 2. «Як вони будуть адмініструватися?». Звичайно, шаблон має бути виражений формально та з використанням мови. Ця альтернатива приводить до чотирьох різних підходів: *логічного; візуалізація; на основі рівнянь; крос-табуляція*. Кожен із цих підходів має історично чіткі математичні корені. Зупинимося на недостатньо описаному в україномовній літературі підході «перехресної табуляції» (Cross Tabulation).

Перехресна табуляція або крос-табуляція (перехресні дані в таблицях) – це основна і дуже проста форма аналізу даних, добре відома в статистиці та широко використовується для звітності. Двовимірна *крос-таблиця (cross-tab*) схожа на електронну таблицю щодо заголовків рядків і стовпців і значень атрибутів. Комірки (cells) в таблиці – це агреговані операції, як правило, кількох значень атрибутів, які зустрічаються разом (со-

occurances). Переважна кількість крос-таблиць фактично є еквівалентом 3D-гістограм, що показують сумісні облікові записи (3D bar graph).

Довірчі мережі, як різновид крос-таблиці, як правило, ілюструються графічним зображенням розподілу ймовірностей (отриманого як результат обчислень). Довірча мережа – це орієнтований граф (directed graph), що складається з вершин (що являють собою змінні) і дуг між вершинами змінних (що являють собою ймовірності).

6.3.2 Аналіз програмного забезпечення Data Mining

Як уже зазначалося, нині на ринку програмних продуктів пропонуються десятки готових до використання систем дейтамайнінгу, деякі з них навіть орієнтовані на широке використання технологічних засобів дейтамайнінгу, а от інші ґрунтуються на специфічних методах (нейромережах, деревах рішень тощо). Охарактеризуємо найновіші системи ДМ з низкою різних підходів і методів дейтамайнінгу − PolyAnalyst, MineSet, Knowlenge STUDIO [20].

PolyAnalyst

Компанія «Мегакомп'ютер» виготовляє та пропонує сімейство продуктів для дейтамайнінгу – PolyAnalyst. Система PolyAnalyst призначена для автоматизованого аналізу числових баз даних і отримання практично корисних знань із звичайних наборів даних. PolyAnalyst шукає багатовимірні зв’язки між змінними в базі даних, автоматично будує та перевіряє багатовимірні нелінійні моделі, що виражають знайдений зв’язок, виводить правила класифікації на основі навчальних прикладів, знаходить багатовимірні кластери в даних і будує алгоритми прийняття рішень.

Нині PolyAnalyst використовується в більш ніж 20 країнах світу для розв'язання задач з різних галузей людської діяльності: бізнесу, фінансів, науки, медицини. Зараз це одна із найпотужніших та доступних за вартістю комерційних систем для виконання дейтамайнінгу.

Робота PolyAnalyst ґрунтується на використанні машин досліджень (Exploration engines), тобто програмні модулі, що ґрунтуються на різних алгоритмах дейтамайнінгу, які призначені для автоматичного аналізу масивів даних.

MineSet -- візуальний інструмент аналітика

Компанія «Silicon Graphics» розробила систему дейтамайнінгу – MineSet, яка відрізняється специфічними особливостями як на концептуальному, так і на технологічному рівнях. Основна увага зосереджена на унікальній процедурі візуальної інтерпретації складних зв’язків у багатовимірних даних.

Система MineSet – це набір інструментів для поглибленого аналізу інтелектуальних даних на основі використання потужної візуальної пара-

дигми. Характерною особливістю MineSet є комплексний підхід, який адаптує використання не однієї, а кількох взаємодоповнювальних стратегій для дослідження, аналізу та інтерпретації даних. Це дає користувачеві можливість вибрати найбільш підходящий інструмент або діапазон інструментів залежно від проблеми, що вирішується, і типу використовуваного обладнання та програмного забезпечення. Архітектура MineSet принципово відкрита – за допомогою стандартизованого формату файлу інші додатки можуть надавати дані для введення в MineSet, а також використовувати результати його роботи. Відкрита архітектура системи також є основою для її майбутнього розширення, яке передбачає можливість включення нових компонентів на основі концепції інтеграції (plug-in). Інтерфейс прикладного програмування (API) дозволяє об'єднувати елементи MineSet в автономні програми [20].

Knowledge STUDIO

Knowledge STUDIO є новою версією дейтамайнінгу корпорації з програмного забезпечення «ANGOSS» (www.angoss.com). Система впроваджує найрозвинутіші методи дейтамайнінгу в корпоративне середовище з тим, щоб підприємства могли досягати максимального прибутку від своїх інвестицій у дані. Ця система забезпечує високу продуктивність користувачів щодо розв'язання ділових проблем без суттєвих затрат на навчання, які потрібні для опанування статистичного програмного забезпечення. Крім того, це також потужний інструментальний засіб для аналітиків.

Knowledge STUDIO сумісна з основними статистичними пакетами програм. Наприклад, ця система не тільки читає та записує файли даних, але й генерує коди статистичних пакетів SAS. Завдяки цим властивостям щодо статистики, розробники таких моделей можуть швидко й легко адаптувати застарілий статистичний аналіз.

Система Knowledge STUDIO тісно інтегрована зі сховищами даних і вікнами. У цьому випадку дані можна досліджувати в режимі інтелектуального аналізу на місці (In-place Mining), тобто коли вони залишаються в сховищі даних «на місці», автоматично використовуючи «хвилі запитів», тобто ряд інструкцій SQL. Оскільки дані надходять безпосередньо з джерела, немає необхідності їх дублювати. Крім того, для оптимізації дейтамайнінгу дані можна вибирати за їх форматом зберігання, а потім інтелектуальний аналіз даних виконується за допомогою високопродуктивного сервера Knowledge STUDIO, орієнтованого на формат файлів.

Технологія ДМ ANGOSS ActiveX інтегрує моделі для прогнозування з Web-базовими додатками і бізнесовими клієнт/серверними додатками. Дослідження даних за допомогою використання дерев рішень та графіки можна розширювати через Інтернет.

Також можна використовувати рішення Java для розгортання моделей. Для виконання обчислювальних алгоритмів у проекті Knowledge STUDIO можна використовувати віддалений «обчислювальний» сервер або локальну робочу станцію.

У Knowledge STUDIO реалізовано величезну кількість методів інтелектуального аналізу даних (дейтамайнінгу). Система пропонує 5 алгоритмів дерев рішень, 3 алгоритми нейронної мережі та алгоритм неконтрольованої кластеризації. Є повна інтеграція з додатками та бізнес-процесами. Можна створювати нові додатки або вставляти дейтамайнінг у наявні. Програмований комплекс Knowledge STUDIO Software (SDK) дозволяє розробляти додатки для створення моделей прогнозування з можливістю використання Power-Builder, Visual Basic, C++, Delphi, Java. Формування, випробування і оцінювання нових моделей може бути також автоматизованим. Knowledge STUDIO забезпечує різні шляхи, щоб візуально виразити і дослідити у великих базах даних зразки прихованих закономірностей [20].

6.4 Технології автоматизованого добування знань з тексту

Технологія глибинного аналізу тексту Text Mining здатна виступати у ролі «викладача», який, вивчивши весь курс, викладає лише найбільш ключову і значущу інформацію, позбавляючи користувача необхідності «просіювати» величезну кількість неструктурованої інформації. Розроблені на основі статистичного і лінгвістичного аналізу та штучного інтелекту технології Text Mining призначені для виконання смислового аналізу, забезпечення навігації і пошуку в неструктурованих текстах. Застосовуючи побудовані на їх основі системи, користувачі можуть отримувати нову цінність − знання. Технологія Text Mining дає змогу аналізувати великі обсяги інформації у пошуках тенденцій, шаблонів і взаємозв'язків, здатних допомогти в прийнятті стратегічних рішень. Крім того, Text Mining − це новий вид пошуку, який, на відміну від традиційних підходів, не лише знаходить списки документів, формально релевантних запитам, але і допомагає зрозуміти зміст досліджуваної проблеми, що передбачає дуже високий рівень інтелектуалізації системи [21].

Текстомайнінг (Text Mining) часто називають також текстовим дейтамайнінгом (Text Data Mining), що частково розкриває взаємозв'язок цих двох технологій. Як Data Mining, так і Text Mining є процесами видобування знань, тобто, за означенням експерта GTE Labs Г. П’ятецького-Шапіро, «процесом виявлення у сирих даних раніше невідомих, нетривіальних, практично корисних і придатних для інтерпретації певних знань, що необхідні для прийняття рішень в різних галузях людської діяльності». Проте, якщо дейтамайнінг дає змогу добувати нові знання (приховані закономірності, факти, невідомі взаємозв'язки тощо) з великих обсягів структурованої інформації, яка

зберігається в базах даних, то текстомайнінг − знаходити нові знання в неструктурованих текстових масивах. Тобто текстомайнінг доповнює технологію дейтамайнінгу додатковим етапом − перетворенням неструктурованих текстових масивів на структуровані, після чого дані можуть оброблятися за допомогою стандартних методів дейтамайнінгу.

Поява технологій глибинного добування даних Data Mining, передувала виникненню технологій глибинного аналізу текстів Text Mining і створила підстави для їх розвитку. Слово «mining» («видобування руди») у назві методів виступає як метафора (виявлення глибоко «заритої» інформації) та символічно вказує на головну особливість цих програмних систем − пошук прихованої інформації і раніше не досліджених закономірностей. Наприкінці XX ст. технологія Text Mining остаточно виділилась у окремий напрям аналізу неструктурованої текстової інформації, який став логічним продовженням Data Mining. Технологія Text Mining поєднала класичні методи добування даних (такі як, наприклад, кластеризація), методи контент-аналізу, статистичного аналізу тощо з реалізацією нових додаткових функцій, таких як автоматичне реферування текстів, виділення понять, феноменів, фактів, зарахування документа до певних категорій згідно із заданою схемою їх систематизації. Сучасні системи Text Mining можуть застосовуватися в управлінні знаннями для виявлення шаблонів у тексті, для автоматичного «виштовхування» чи розміщення інформації за профілями, що цікавлять користувачів, створення оглядів документів.

Одна з найважливіших компонент технології Text Mining пов'язана з добуванням із тексту його ключових слів, анотацій інших характерних елементів чи властивостей, які можуть використовуватися як метадані документа. Text Mining також забезпечує новий рівень семантичного пошуку документів та їх зарахування до певних категорій заданої схеми систематизації документів

Отже, Text Mining можна визначити як «алгоритмічне виявлення на основі систем штучного інтелекту, статистичного і лінгвістичного аналізу раніше невідомих зв'язків і кореляцій у вже існуючих неструктурованих текстових даних для проведення смислового аналізу, забезпечення навігації і пошуку в неструктурованих текстах з кінцевою метою отримання нової цінної інформації − знань». Відмінність технології Text Mining від Data Mining полягає в тому, що остання працює з базами даних, тобто зі структурованою інформацією, тоді як Text Mining дає дослідникові змогу аналізувати неструктуровану інформацію, подану в формі звичайних текстів на природній мові [21].

Актуальність текстомайнінгу (зростає у міру того, як людям різних професій доводиться приймати рішення на базі аналізу великого обсягу неструктурованих і слабкоструктурованих текстів, частка яких у вигляді текстових файлів, файлів електронної пошти, систем управління контентом, систем управління документами, онтологій, таксономій, Web-

сторінок, мультимедійних файлів тощо останніми роками зростає на 5÷8 % за рік.

Умовно систему текстомайнінгу можна розділити на чотири блоки (рис. 6.4) [21].

Рисунок 6.4 − Структура системи текстомайнінгу

Модуль передпроцесингу об'єднує технології добування і фільтрації текстів, що надходять на обробку. Модуль штучного інтелекту відповідає за «розуміння» текстів природною мовою. Наступний модуль містить засоби підтримки реалізації набору необхідних користувачеві задач, кожна з яких потребує певного технологічного розв'язання.

У загальному випадку модуль реалізації завдань користувача може підтримувати досить великий набір задач, зокрема [21]:

- класифікацію;
- кластеризацію;
- побудову семантичних мереж;
- добування фактів, понять (feature extraction);
- добування думок;
- анотування, реферування, формування огляду (summarization),
- формування відповіді на запит (question answering);
- тематичне індексування (thematic indexing);
- пошук за ключовими словами (keyword searching);
- створення таксономій (ієрархічних деревоподібних класифікацій) та тезаурусів.

Відповідно до цих завдань до основних інструментів Text Mining зараховують засоби анотування чи побудови резюме (реферату)

(summarization), виділення феноменів, властивостей та понять (feature extraction), кластеризації (clustering), класифікації (classification), формування відповіді на запити (question answering), тематичного індексування (thematic indexing) і пошуку за ключовими словами (keyword searching). Також у деяких системах Text Mining набір інструментів доповнюють засоби підтримки і створення таксономії (taxonomies) і тезаурусів (thesauri). Деякі фахівці виділяють чотири основні завдання, розв'язувані технологією Text Mining:

- *класифікація* тексту на основі побудови правил зарахування документів до завчасно визначених категорій за результатами виявлення статистичних кореляцій між елементами тексту та елементами шаблону з бази шаблонів системи;
- *кластеризація* за допомогою лінгвістичних та математичних методів виявлення та обробки ознак документів без використання зумовлених категорій з отриманням як результат таксономії чи візуальної карти, яка забезпечує ефективне охоплення великих обсягів даних;
- *побудова семантичних мереж* та аналіз зв'язків, які визначають появу дескрипторів (ключових слів чи фраз) у документі для забезпечення навігації;
- *видобування фактів*, призначене для одержання з тексту певних фактів з метою покращання класифікації, пошуку і кластеризації.

Всі системи Text Mining містять інструменти розв'язання задачі класифікації, яку вважають найпоширенішою задачею глибинного аналізу текстів. Класифікація застосовується, наприклад, під час групування документів Intranet-мережах та на Web-сайтах, розміщенні документів у певні папки, сортуванні повідомлень електронної пошти, вибірковому поширенні новин передплатникам тощо. На практиці задача класифікації зводиться до класичної задачі розпізнавання, де за навчальною вибіркою система зараховує новий об'єкт до тієї чи іншої категорії. Особливість текстомайнінгових систем полягає в тому, що кількість об'єктів і їх атрибутів може бути дуже великою; тому потрібно передбачити інтелектуальні механізми оптимізації процесу класифікації [21].

Кластеризація застосовується під час реферування великих документальних масивів, визначення взаємопов'язаних груп документів, для спрощення процесу перегляду під час пошуку необхідної інформації, знаходження унікальних документів з колекції, виявлення дублікатів або дуже близьких за змістом документів. Розв'язуючи задачу кластеризації, тобто виділення компактних підгруп об'єктів із близькими властивостями, система має самостійно знайти ознаки і розділити об'єкти на підгрупи. Кластеризація переважно передує класифікації, оскільки дає змогу визначити групи (категорії) об'єктів. У текстомайнінгових системах застосовують два основні типи кластеризації − ієрархічну та бінарну. Ієрархічна кластеризація полягає в побудові дерева кластерів, в кожному з яких розміщується

невелика група документів. Двійкова кластеризація забезпечує групування і переглядання документальних кластерів за посиланнями подібності. До одного кластера входять найближчі за властивостями документи. В процесі кластеризації будується базис посилань від документа до документа, оснований на вагах і спільному вживанні ключових слів, які визначаються в процесі аналізу тексту.

Технологія текстомайнінгу дає змогу також розв'язувати задачі прогнозування, тобто передбачення за значенням одних ознак об'єкта значень всіх інших, та задачі знаходження винятків, тобто пошуку об'єктів, які за характеристиками суттєво виділяються із загальної множини об'єктів. Для цього спочатку з'ясовують середні параметри об'єктів, а потім досліджують ті об'єкти, параметри яких найбільше відрізняються від середніх значень. Пошук винятків поширений, наприклад, в роботі спецслужб. Такий аналіз часто проводять після класифікації, щоб оцінити точність останньої.

Окрім кластеризації, текстомайнінг підтримує пошук зв'язаних ознак (полів, понять) окремих документів. Ці ознаки є такими самими, як і за кластеризації. Від передбачення задача такого пошуку відрізняється тим, що наперед не відомо, за якими саме ознаками реалізується взаємозв'язок; мета полягає саме у знаходженні зв'язків ознак.

Один із модулів текстомайнінгової системи (див. рис. 6.4) – модуль інтерфейсу. Модуль містить засоби, які формують графічний інтерфейс користувача, і забезпечують належне подання інформації, що дає людині змогу побачити додаткові приховані закономірності, які не вдається виявити іншими методами. Візуалізація має велике значення для обробки й інтерпретації результатів текстомайнінгу. Візуалізація даних передбачає обробку структурованих числових даних, проте водночас відіграє ключову роль у поданні схем неструктурованих текстових документів. Зокрема, сучасні текстомайнінгові системи здатні аналізувати великі масиви документів і формувати наочні покажчики понять та тем, висвітлених у цих документах. Візуалізація зазвичай використовується як засіб подання змісту (контенту) всього масиву документів, а також для реалізації навігаційного механізму, який може застосовуватися під час дослідження документів і їх класів.

Сьогодні існує досить потужне програмне забезпечення, що реалізує методи Text Mining. Переважно це масштабовані системи, з розвиненими графічними інтерфейсами, великими можливостями з візуалізації і маніпулювання даними, що надають доступ до різних джерел даних та функціонують в архітектурі клієнт-сервер. Ці системи відповідають сучасним вимогам як за архітектурою, так і за функціональними можливостями. Вони подані як порівняно простими програмами, що спираються на статистичний аналіз окремих термінів у текстах (такі як WordStat), так і надскладними додатками типу Aerotext та Business Objects Text Analysis [21].

6.5 Інтелектуальний аналіз даних в СУБД Microsoft SQL Server

Розглянемо реалізацію засобів інтелектуального аналізу даних в СУБД Microsoft SQL Server. Ці задачі розв'язуються службами Analysis Services [24].

Служби Analysis Services надають такі функції і засоби для створення рішень з інтелектуального аналізу даних:

1. Набір стандартних алгоритмів інтелектуального аналізу даних;
2. Конструктор інтелектуального аналізу даних, призначений для створення і перегляду моделей інтелектуального аналізу даних, управління ними та побудови прогнозів;
3. Мову розширень інтелектуального аналізу даних (Data Mininge Xtensionsto SQL, DMX).

Для роботи із наданими засобами інтелектуального аналізу використовується середовище Business Intelligence Development Studio, скорочено BIDevStudio (рис. 6.5, 6.6).

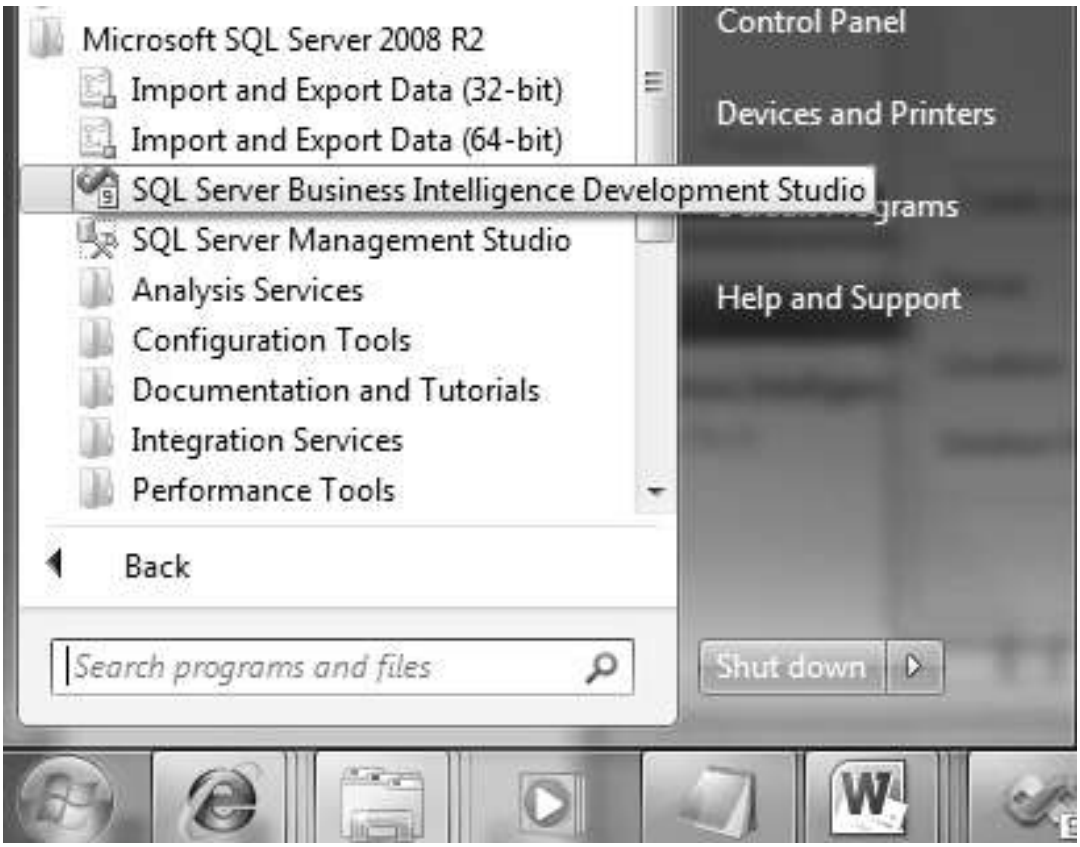


Рисунок 6.5 − Запуск SQL Server Business Intelligence Development Studio

Також SQL Server підтримують створення, управління і використання моделей інтелектуального аналізу даних з Microsoft Excel за допомогою надбудов інтелектуального аналізу даних SQL Server для Office.

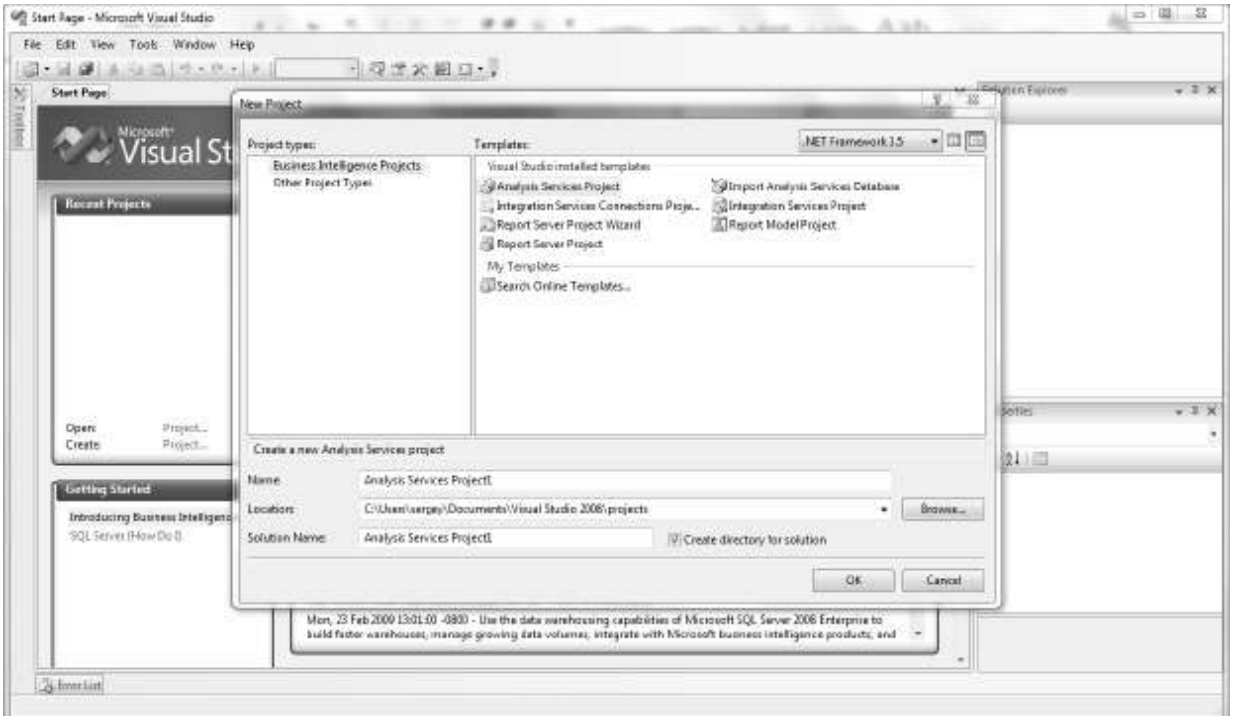


Рисунок 6.6 − Створення нового проекту в Business Intelligence Development Studio

Структура інтелектуального аналізу даних може бути подана як сукупність вихідних даних і опису способів їх обробки. Структура містить моделі, які використовуються для аналізу її даних. Зокрема, одна структура може підтримувати кілька моделей. У структурі інтелектуального аналізу даних можна виділити набір даних як для навчання, так і для перевірки, задавши процентне відношення або обсяг даних.

Модель інтелектуального аналізу даних є поєднанням самих даних, алгоритму інтелектуального аналізу даних і колекції значень параметрів та фільтрів, які керують використанням і обробкою даних. Модель інтелектуального аналізу даних визначається мовою розширень інтелектуального аналізу даних або за допомогою майстра інтелектуального аналізу даних в середовищі BI DevStudio.

Алгоритм інтелектуального аналізу даних є механізм, що створює модель інтелектуального аналізу даних. Щоб створити модель, алгоритм спочатку аналізує набір даних, здійснюючи пошук певних закономірностей і трендів. Алгоритм використовує результати цього аналізу для визначення параметрів моделі інтелектуального аналізу даних. Потім ці параметри застосовуються до всього набору даних, щоб виявити придатні до використання закономірності і отримати докладну статистику.

Нижче перераховано алгоритми інтелектуального аналізу даних, реалізовані в Microsoft SQL Server (вказівка на Microsoft говорить про те, що це її реалізації алгоритмів) [24]:

1. Спрощений алгоритм Байєса – MicrosoftNaiveBayes;
2. Алгоритм дерева прийняття рішень – MicrosoftDecisionTrees;

3. Алгоритм часових рядів – MicrosoftTimeSeries;

4. Алгоритм кластеризації – MicrosoftClustering;

5. Алгоритм кластеризації послідовностей – Microsoft Sequence Clustering;

6. Алгоритм взаємозв'язків – Microsoft Association Rules;

7. Алгоритм нейронної мережі – Microsoft NeuralNetwork;

8. Алгоритм лінійної регресії – MicrosoftLinearRegression;

9. Алгоритм логістичної регресії – MicrosoftLogisticRegression.

В табл. 6.2 наведено деякі приклади використання інтелектуального аналізу даних і відповідні їм алгоритми.

Таблиця 6.2 − Приклади використання алгоритмів інтелектуального аналізу

Задачі та приклад	**Відповідні алгоритми**
Прогнозування дискретного атрибута. *Наприклад, чи купить одержувач цільової розсилки певний продукт.*	Алгоритм дерева прийняття рішень. Спрощений алгоритм Байєса. Алгоритм кластеризації. Алгоритм нейронної мережі
Прогнозування безперервного атрибута. *Наприклад, прогноз продажів на наступний рік.*	Алгоритм дерева прийняття рішень. Алгоритм тимчасових рядів.
Прогнозування послідовності. *Наприклад, аналіз маршруту переміщення по веб-сайту компанії*	Алгоритм кластеризації послідовностей
Знаходження груп загальних елементів в транзакціях. *Наприклад, використання аналізу купівельної поведінки для пропозиції додаткових продуктів замовнику.*	Алгоритм взаємозв'язків Алгоритм дерева прийняття рішень
Знаходження груп схожих елементів. *Наприклад, розбиття демографічних даних на групи для кращого розуміння зв'язків між атрибутами.*	Алгоритм кластеризації Алгоритм кластеризації послідовностей

Питання для самоконтролю

1. Розкрийте суть технології інтерактивної аналітичної обробки даних (OLAP).
2. Розкрийте суть DM та охарактеризуйте його методи.
3. Охарактеризуйте етапи проведення DM.
4. Розкрийте суть DM та охарактеризуйте його моделі.
5. Охарактеризуйте програмне забезпечення DM.
6. Розкрийте суть Text Mining та охарактеризуйте його структуру.
7. Наведіть та охарактеризуйте основні завдання, розв'язувані технологією Text Mining.
8. Розкрийте суть інтелектуального аналізу даних в СУБД Microsoft SQL Server.
9. Наведіть алгоритми інтелектуального аналізу даних, які реалізовані в Microsoft SQL Server.
10. Наведіть приклади використання алгоритмів інтелектуального аналізу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кун Томас. Структура наукових революцій [Електронний ресурс]. – Режим доступу: http://litopys.org.ua/kuhn/kuhn.htm (дата звернення 12.08.2023). – Назва з екрана.

2. Надобко С. В. Навчально-методичний посібник з дисципліни «Методика організації науково-дослідної роботи» для студентів усіх спеціальностей денної та заочної форми навчання / Надобко С. В. − Харків : ХДАДМ, 2020. – 125 с.

3. Парадигма (мовознавство) [Електронний ресурс]. − Режим доступу: https://uk.unionpedia.org/i/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0_(%D0%BC%D0%BE%D0%B2%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D1%81%D1%82%D0%B2%D0%BE) (дата звернення 12.08.2023). – Назва з екрана.

4. Пардигма – Словник української мови у 20 томах [Електронний ресурс]. − Режим доступу: https://slovnyk.me/dict/newsum/%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0 (дата звернення 12.08.2023). – Назва з екрана.

5. Парадигми програмування – Словник з інформатики [Електронний ресурс]. − Режим доступу: http://xn--r1a3b.xn--b1amgblet.xn--j1amh/index.php/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення 12.08.2022). – Назва з екрана.

6. Інформаційно-технологічна парадигма [Електронний ресурс]. − Режим доступу :

https://stud.com.ua/48382/politekonomiya/informatsiyno_tehnologichna_paradigma (дата звернення 12.08.2023). – Назва з екрана.

7. Шмідт Е. Новий цифровий світ. Як технології змінюють державу, бізнес і наше життя / Е. Шмідт, Дж. Коен. – Львів : Літопис, 2015. – 368 с.

8. Огляд «The New Digital Age: Reshaping the Future of People, Nations and Business» [Електронний ресурс]. – Режим доступу: https://hub.kyivstar.ua/reviews/novyj-czyfrovyj-svit/ (дата звернення 12.08.2022). – Назва з екрана.

9. Топольник В. Г. Метрологія, стандартизація, сертифікація і управління якістю : навч. посіб. / В. Г. Топольник, М. А. Котляр. – Львів : Магнолія–2006, 2009. – 212 с.

10. Саранча Г. А. Метрологія, стандартизація, відповідність, акредитація та управління якістю : підруч. / Г. А. Саранча. – К. : Центр навч. літератури, 2006. – 672 с.

11. Міжнародні структури в галузі стандартизації інформаційних технологій [Електронний ресурс]. − Режим доступу:

https://stud.com.ua/35757/informatika/mizhnarodni_strukturi_galuzi_standartizatsiyi_informatsiynih_tehnologiy (дата звернення 12.08.2023). – Назва з екрана.

12. Яровий А. А. Методи та засоби організації високопродуктивних паралельно-ієрархічних обчислювальних систем із рекурсивною архітектурою : монографія / Яровий А. А. – Вінниця : ВНТУ, 2016. – 363 с.

13. Тарарака В. Д. Архітектура комп'ютерних систем: навч. посіб. / Тарарака В. Д. – Житомир : ЖДТУ, 2018. – 383 с.

14. Погорілий С. Д. Новітні архітектури відеоадаптерів. Технологія GPGPU. Частина 1 / С. Д. Погорілий, Д. Ю. Вітель, О. А. Верещинський // Реєстрація, зберігання і обробка даних. Серія: Технічні засоби отримання і обробки даних. – 2012. − Т 14, №4. − С. 53−64.

15. Haykin S. Neural networks : a comprehensive foundation / S. Haykin. – second edition. – Upper Saddle River, New Jersey 07458 : Prentice Hall, Inc, 1999. − 1104 p.

16. Програми моделювання штучних нейронних мереж [Електронний ресурс]. − Режим доступу: http://um.co.ua/1/1-1/1-12540.html (дата звернення 12.08.2023). – Назва з екрана.

17. Методичні вказівки до виконання лабораторних робіт з дисципліни «Технологія та використання штучних нейронних мереж» для студентів напряму підготовки 6.050103 «Програмна інженерія» усіх форм навчання / Уклад. : С. О. Субботін, Є. М. Федорченко. – Запоріжжя: ЗНТУ, 2013. – 60 с.

18. Бориченко О. В. Використання програмного пакета MATLAB для побудови штучних нейронних мереж / О. В. Бориченко, О. Б. Іванець, О. В. Букрєєва // Електроніка та системи управління. − 2011. − №3(29). – С. 120−123.

19. Інтерфейс мозок-комп'ютер: від відкриття до впровадження [Електронний ресурс]. − Режим доступу: https://phm.cuspu.edu.ua/nauka/naukovo-populiarni-publikatsii/1004-interfeis-mozok-komp-iuter-vid-vidkryttia-do-vprovadzhennia.html (дата звернення 12.08.2023). – Назва з екрана.

20. Ситник В. Ф. Системи підтримки прийняття рішень : навч. посіб. / Ситник В. Ф. − К.: КНЕУ, 2004. − 614 с.

21. Павлиш В. А. Основи інформаційних технологій і систем : навч. посіб. / В. А. Павлиш, Л. К. Гліненко. – Львів : Видавництво Львівської політехніки, 2013. – 500 с.

22. Месюра В. І. Експертні системи : навч. посіб. : у 2 ч. / Месюра В. І., Яровий А. А., Арсенюк І. Р. – Вінниця : ВНТУ, 2006. – Ч. 1. − 114 с.

23. Яровий А. А. Експертні системи : навч. посіб. : у 2 ч. / Яровий А. А., Арсенюк І. Р., Месюра В. І. – Вінниця : ВНТУ, 2017. – Ч. 2. – 106 с.

24. Лосєв М. Ю. Бази даних : навчально-практичний посібник для самостійної роботи студентів [Електронний ресурс] / М. Ю. Лосєв, В. В. Федько. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 233 с.

Навчальне електронне видання
комбінованого використання.
Можна використовувати в локальному та мережному режимах

Андрій Анатолійович Яровий
Людмила Вікторівна Крилик
Андрій Володимирович Козловський

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У СФЕРІ ШТУЧНОГО ІНТЕЛЕКТУ

Навчальний посібник

Рукопис оформила *Л. Крилик*

Редактор *Т. Старічек*

Оригінал-макет підготувала *Т. Старічек*

Підписано до видання 18.10.2023 р.
Гарнітура Times New Roman.
Зам. № Р2023-122.

Видавець та виготовлювач
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95, м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
E-mail: irvc.ed.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.