

## АРХІТЕКТУРА ОПРАЦЮВАННЯ ЗНАНЬ У КОГНІТИВНІЙ ІНФОРМАЦІЙНІЙ СИСТЕМІ

© Буров Є., 2009

**Запропоновано підхід та архітектуру подання та опрацювання знань як основу для побудови когнітивних інформаційних систем. За цим підходом поєднуються декларативні онтології та процедурні моделі. Наведено формальну специфікацію архітектури опрацювання знань у системі.**

**Paper proposes an approach for knowledge representation and processing which can be used as a basis for building cognitive information systems. Proposed architecture combines ontologies as declarative knowledge with models as procedural and situational knowledge. Formal specification of proposed knowledge representation architecture is available.**

### Постановка проблеми та мета роботи

Зростання складності інформаційних та технічних систем, збільшення ступеня інтеграції бізнес-процесів (БП) підприємств, розвиток електронної комерції та глобалізація економіки вимагають, з одного боку, поглиблення рівня аналізу БП, а з іншого – швидкої реакції на зміни бізнес-середовища. Актуальними напрямками досліджень для вирішення цих задач є видобування знань про БП (Process Mining) [1] та бізнес-аналітика (Business Intelligence). Зокрема одним з головних завдань другої генерації бізнес-аналітики (BI 2.0) є швидке реагування на події, що виникають у бізнесовому середовищі [2].

Шляхом вирішення цих завдань є подальша автоматизація виконання БП та запровадження систем та компонент штучного інтелекту для керування та аналізу БП. Отже, одним з напрямів розвитку систем моделювання БП є перехід від переважно ручних (або з використанням допоміжних програмних засобів) методів аналізу та керування БП до переважно автоматичного керування виконанням БП, побудови когнітивних мереж БП у вигляді набору взаємопов'язаних моделей, які інкапсулюють знання про структуру, особливості БП, системні події, обмеження та залежності та опрацьовуються машинно. При цьому такій системі делегуються повноваження з прийняття рішень у чітко визначених (найчастіше простих, рутинних) ситуаціях.

У роботі [3] запропоновано архітектуру та формальну специфікацію інтелектуальної системи моделювання БП як основи для побудови когнітивних інформаційних систем. Водночас відкритим залишилося питання визначення вимог до когнітивної інформаційної системи, дослідження методів представлення знань в ній, принципів їх використання.

**Метою** цієї роботи є побудова робочого визначення когнітивної інформаційної системи та вимог до неї, розроблення підходу та архітектури представлення та опрацювання знань у такій системі.

### Когнітивні системи та існуючі підходи до їх аналізу та створення

Термін “Когнітивність” (Cognition, від латинського *cognoscere* – “знати”) використовується у різноманітних науках та контекстах для позначення здатності людиноподібного опрацювання інформації, застосування знань та динамічної зміни переваг і поведінки.

Словникові та енциклопедичні джерела дають таке визначення когнітивності:

“Внутрішні структури та процеси, що задіяні у здобутті та використанні знань, включаючи відчуття, сприйняття, увагу, навчання, пам'ять, розуміння мови, думання та розумування” [4].

Отже, поняття когнітивності об'єднує широкий набір інтелектуальних властивостей, таких як сприйняття інформації, її розпізнавання та класифікація, навчання, розумування, накопичування, збереження та використання знань. Тобто термін „когнітивність”, когнітивні системи, так як і загальніший термін 'Загальний Інтелект' (Artificial General Intelligence), за образним визначенням одного з класиків штучного інтелекту Марвіна Мінського, є терміном-контейнером (suitcase) [5], що охоплює багато складних аспектів та проблем, які у комплексі формують цілісну систему та можуть вивчатися окремо.

На думку багатьох дослідників [6], неможливо дати єдине правильне визначення подібних фундаментальних понять. Тому кожний дослідник, як правило, формулює своє робоче визначення, яке охоплює важливі, на його думку, аспекти досліджуваного поняття і яке врешті-решт визначає як мету, так і результати дослідження.

Об'єктом дослідження у нашій роботі є когнітивні інформаційні системи, тобто інформаційні системи, наділені когнітивними функціями.

Наукова галузь штучного інтелекту з самого початку свого існування приділяла значну увагу вивченню питання як когнітивних систем загалом, так і окремих аспектів цих систем.

У класичній роботі „Уніфікована теорія когнітивності” (Unified theory of cognition) [7] Ньювелл (Newell) стверджував, що наявні експериментальні результати у галузі дослідження когнітивної поведінки людини (експериментальна психологія) дають змогу побудувати уніфіковану теорію пізнання. Така теорія пропонувала би механізми відтворення результатів когнітивної поведінки людей.

У [8] наголошується, що дослідження у галузі штучного інтелекту сьогодні зосередилися у двох напрямках: у першому розглядаються загальні теоретичні питання штучного інтелекту, не пов'язані з практикою створення інтелектуальних систем. Другий напрямок розглядає конкретні задачі застосування систем штучного інтелекту у вузьких проблемних областях (наприклад, гра в шахи, водіння авто).

При цьому результати, отримані для розв'язання вузьких задач, не завжди можуть бути застосовані для розв'язання задач в інших проблемних областях. Не завжди також їх можна застосувати для розв'язання загальних задач, подібно до того, як ці задачі вирішує людина.

Сьогодні сформувалося два підходи до вирішення питання отримання знань у процесі побудови загальних когнітивних систем. Класичний підхід використовує базу знань з фіксованою схемою. Знання в такій системі привносяться ззовні, програмуються.

За підходом, що ґрунтується на паттернах [9], приймається, що розум є множиною паттернів (конфігурацій, образів) у мозку. Паттерни як структури неможливо запрограмувати наперед, вони повинні самі сформуватися самі у процесі розвитку системи під час інтерактивного навчання, як досвід такого навчання.

Класичний підхід зокрема використовується у робототехніці і при побудові інтелектуальних агентів. Сильною стороною класичного підходу є можливість документування та програмування знань та поведінки системи. Це робить можливим навчання системи шляхом прямого формулювання правил, а також полегшує контроль за змістом бази знань. Недоліком є складність адаптації до непередбачених обставин, неможливість самонавчання та саморозвитку.

Прикладом архітектури, яка ґрунтується на паттернах, є система Novamente [9]. Особливістю є використання інтегративного підходу – комбінація традиційного подання знань з роботою декількох когнітивних процесів, що працюють з цим знанням. У результаті формується автономна, адаптивна система, що навчається на своєму досвіді і в якій у результаті взаємодії когнітивних процесів виникають риси та можливості загального інтелекту.

Істотною перевагою системи Novamente є здатність до самонавчання та самостійного формування понять. До недоліків належать складність системи, складності в інтеграції існуючих знань у систему та пояснення новонабутих знань. Останнє пов'язане з тим, що динамічні паттерни, подані як карти знань, доволі складно відобразити у формі, придатній для сприйняття людиною, і, наприклад, для подальших перевірки та валідації. Водночас, значний обсяг накопичених знань існує у формі, придатній для сприйняття людиною. Як правило, такі знання просто закодувати з

використанням класичного підходу, який ґрунтується на використанні правил, символічних моделей та логіки. Водночас доволі складно створити таку процедуру навчання системи, орієнтованої на паттерни, щоб у результаті навчання сформувалися визначені існуючі знання.

Підхід до побудови когнітивних систем, поданий у [10] має на меті побудову людино-машинних систем. Когнітивна машинна система тут взаємодіє з людиною-оператором та утворює з нею єдине ціле. У роботі дано визначення когнітивної системи, вимоги та властивості такої системи, методи та стратегії боротьби зі складністю та недостатньою інформацією в людино-машинних системах.

Узагальнюючи функції та вимоги до когнітивних систем з [5–10], сформулюємо головні риси робочого визначення когнітивної системи:

1. Когнітивна система має мету функціонування. Ця мета часто розбивається на ієрархічну структуру цілей. При цьому важливість тієї чи іншої підцілі залежить від поточного контексту.

2. Система адаптивна та здатна приймати рішення залежно від власного стану та стану середовища.

3. Працює з використанням знань про себе та своє оточення та здатна планувати та модифікувати свої дії на підставі цих знань.

4. Здатна навчатися як із зовнішнім вчителем, так і самостійно.

5. Система використовує інформацію зворотного зв'язку для прийняття рішення.

6. Прогнозує результати можливих дій і приймає рішення за результатами прогнозів.

7. Враховує поточний контекст у процесі прийняття рішення.

8. Працює на основі знань, які мають семантичну інтерпретацію.

9. Співпрацює з людиною, тобто правильність її роботи може перевіряти та оцінювати людина. Сприймає та передає знання у формі, придатній для розуміння людиною.

10. Здатна співпрацювати з іншою когнітивною системою та обмінюватися з нею знаннями.

Завдання побудови когнітивних інформаційних систем особливо актуальна в сучасних умовах, коли поява інтелектуальних програмних агентів вимагає від них обміну знаннями не тільки з людьми, але й з іншими інтелектуальними агентами.

Необхідною передумовою побудови когнітивних систем для керування та проектування інформаційних систем є розроблення та використання методів і засобів подання та використання знань – складових елементів когнітивних систем.

### **Методи представлення знань у когнітивних інформаційних системах**

Результати досліджень у галузі когнітивної психології [11,12], зокрема теорія схем, істотно вплинули на розвиток теорії систем штучного інтелекту. Найближчим до теорії схем підходом була запропонована М. Мінські у 1975 році [13] теорія фреймів як головний засіб для подання знань про певну предметну область.

Бачення фреймових систем, сформульованих Мінські, було розвинуто у пізніших роботах, зокрема було створено розвинуті фреймові системи KRL та FRL [14]

Існуючі фреймові системи [12] мають такі риси:

- фрейми, організовані у набір пов'язаних ієрархій;
- фрейми складаються зі слотів (атрибутів), для яких визначено «заповнювачі» – скалярні значення, або посилання на інші фрейми або процедури, що дають змогу визначити значення слоту;
- властивості успадковуються фреймами ієрархії відповідно до певної стратегії успадкування.

Ці принципи виявилися дуже корисними на практиці, зокрема вони були покладені в основу сучасних об'єктних мов програмування, об'єктного підходу до проектування систем.

Значна частина розробок в галузі подання знань у системах штучного інтелекту зосередилася на вивченні представлення декларативних знань у фреймових системах. Зокрема були запропоновані онтології як декларативні формальні моделі предметної області [15].

Перевагами декларативних, мережевих систем подання знань є можливість створення цілісної формальної моделі предметної області, досягнення однозначного трактування концептів предметної

області у різних задачах, ідентифікація та формальне представлення простих залежностей між концептами. Водночас декларативні системи подання знань та пов'язані з ними механізми логічного виводу мають і такі недоліки:

- велика складність самої мережевої моделі (онтології) та пов'язані з цим значні витрати на створення та супровід декларативних онтологічних систем;
- процедурна складова у мережевих моделях мають другорядне значення та доволі спрощена;
- непридатність до представлення знань, орієнтованих на розв'язання конкретних задач, відсутність мети;
- непристосованість до подання параметричних, ситуаційних, суб'єктивних, аксіологічних знань, сенсорної інформації;
- непристосованість до подання та використання процедурних абстракцій;
- онтологічні мережеві моделі використовують апарат логічного виводу для отримання результатів моделювання і, отже не можуть містити суперечливих, або несумісних логічних правил та обмежень.

Альтернативою до декларативного підходу до представлення знань є методологія проектування інтелектуальних систем, основана на схемах (SBD – Schema based design) [16].

Головними відмінностями схемного (SBD) підходу до подання та опрацювання знань від мережевого є такі:

- схема описує фрагмент світу і релевантна тільки у певних ситуаціях. Отже, говорять, що схема містить ситуативні знання. При виникненні релевантної ситуації схема активізується;
- схема, як правило, призначена для розв'язання певної задачі та досягнення певної мети;
- порівняно з онтологією кожна схема є відносно простою. Натомість у процесі вирішення інтелектуальних задач формуються агрегати активованих схем, що у комплексі дають змогу розв'язувати складні задачі;
- на відміну від онтологій, схеми не утворюють однієї загальної ієрархії. Зв'язки між схемами часто є асоціативними, горизонтальними. Вони часто змінюються;
- допустиме співіснування суперечливих схем, що активізуються у різних ситуаціях;
- схеми дають змогу подати та використати у розумуванні абстрактні поняття, такі як стереотипи, соціальні ролі, архетипи та ін.;
- схеми об'єднують декларативні та процедурні знання;
- схеми SBD подібні до схем, що використовуються у когнітивній психології і тому будуть зрозумілими як людині, так і машині, що є однією з вимог прийнятого нами робочого визначення штучних когнітивних систем.

Центральним у теорії схем (SBD) є не стільки що може зробити окрема схема, але як вони організовані і що вони можуть зробити разом. Таке бачення було покладено в основу багатьох досліджень у когнітивній психології, штучному інтелекті та когнітивній робототехніці. Були запропоновані подібні до схем структури, такі як фрейми (Minsky, 1975), скріпти (Schank and Abelson, 1977), схеми (Arbib, 1992), нейронні схеми (McCaughey, 2002), схеми семіотики (Roy, 2005), поведінки (Brooks, 1991; Maes, 1990) та ін. [12]

На відміну від онтологій, моделі та схеми як форми представлення знань мало досліджені. Увага при цьому приділяється використанню схем у робототехніці (переважно перцептивних та моторних схем), моделюванні структур людського мозку (нейронні схеми) тощо. Проблематика використання підходу SBD до проектування інформаційних систем практично не досліджена. Не вирішено велике коло проблем, пов'язаних з організацією розумування та прийняття рішень з використанням моделей, збереження та передача знань, поданих у схемах. Не достатньо досліджені і питання організації навчання на схемах, проблеми зміни та валідації схем.

Недоліками підходу, орієнтованого на схеми, є значна складність побудови систем SBD, що значною мірою впливає з слабозв'язаності окремих схем та постійної їх зміни.

Поняття схеми використовується у різних контекстах та для розв'язання різних задач. Схеми для опрацювання стимулів та контролю рецепторного сприйняття часто називають перцепційними (perceptual) схемами, а схеми, що керують переміщенням та рухами – моторними схемами [17].

За ознакою наявності передбачення результатів у схемі їх поділяють на реактивні (передбачення відсутнє) та з передбаченням (anticipatory). У початковому розумінні схеми, які описували поведінку, всі були реактивними. Сьогодні такі схеми є де-факто стандартом для автономних роботів.

Схеми класифікують і за типами знань. Так, вчені, що займаються когнітивними системами, визначають такі аспекти знання про домен: концептуальні знання (логічне подання концептів області та семантичних зв'язків між ними), епізодичне знання (ментальне подання аудіо-візуального сприйняття реальних об'єктів, ситуацій, подій), подання за аналогією (ментальні моделі, зображення, що зберігають структуру реального об'єкта в аналогії), процедурне знання ( подане як пари “умова–дія”), знання, пов'язане з діями, які необхідно виконати (enactive knowledge), ситуаційне знання (знання у соціально-культурному контексті щоденної діяльності) [18].

Інформаційні системи як об'єкт проектування та дослідження істотно відрізняються від робототехнічних систем, які зосереджуються переважно на перцепційних та моторних схемах та задачах. Моделі та схеми, які застосовуються на різних рівнях запропонованої трирівневої ієрархії моделювання інформаційних систем [19], та вимоги до них також відрізняються.

Так, на рівні бізнес процесів моделі відображають бізнес-логіку, корпоративні політики та обмеження. Вимоги щодо часу реакції є змінними та залежать від конкретної події, але загалом обмеження за часом реагування є менш жорсткими, ніж на інших рівнях моделювання. Це пов'язано з тим, що у бізнесовому середовищі існують довготривалі бізнес-процеси, які розгортаються протягом днів, нерідко – місяців або років. Натомість важливо передбачити результати прийняття рішення та попередньо оцінити витрати, адже вартість можливої помилки може бути високою. Важливою вимогою є координація процесів, оптимізація використання спільних ресурсів.

На рівні сервісів, крім моделей виконання запитів, важливими є моделі оцінки та підтримки якості сервісу. На рівні пристроїв важливо оцінювати працездатність та «здоров'я» пристрою, що вимагає побудови та використання відповідних моделей.

Загалом, на всіх рівнях моделювання важливо передбачити наслідки, координувати використання спільних ресурсів, оцінювати вартість альтернативних варіантів досягнення мети.

### **Архітектура опрацювання знань у когнітивній інформаційній системі**

Запропонована система опрацювання знань та моделювання складається з таких компонент (рис. 1).

База знань містить факти про об'єкти та події зовнішнього світу, необхідні для розв'язання задач системою. Всі факти семантично інтерпретовані, тобто подані як об'єкти певних класів, визначених онтологією. Інформація в базі знань впорядкована за часом та змінами, що дає змогу відслідкувати стан бази на довільний момент часу або на довільну зміну.

База знань також має посилання на зовнішні джерела, наприклад, бази даних, репозиторії документів, веб-сервіси та ін. Важливим зовнішнім джерелом для надходження інформації у систему є коментарі та зміни, внесені вчителем (ментором).

Онтологія містить модель предметної області, подану як ієрархія класів. Це створює можливість однозначного трактування усіх об'єктів з бази знань, визначення єдиної структури слотів та типів властивостей. Крім того, вона містить відношення, правила та обмеження, які мають загальний, системний характер.

Схеми та моделі відіграють багато ролей та мають багато функцій. Для термінологічного розділення цих двох близьких понять будемо використовувати термін «схема моделі» (або просто «схема») для позначення структури та інтерфейсу моделі, який видимий іншими моделями у системі та використовується для керування моделлю та її розуміння, а термін «модель» – для позначення загальнішого поняття, яке передбачає, крім схеми, і реалізацію моделі.

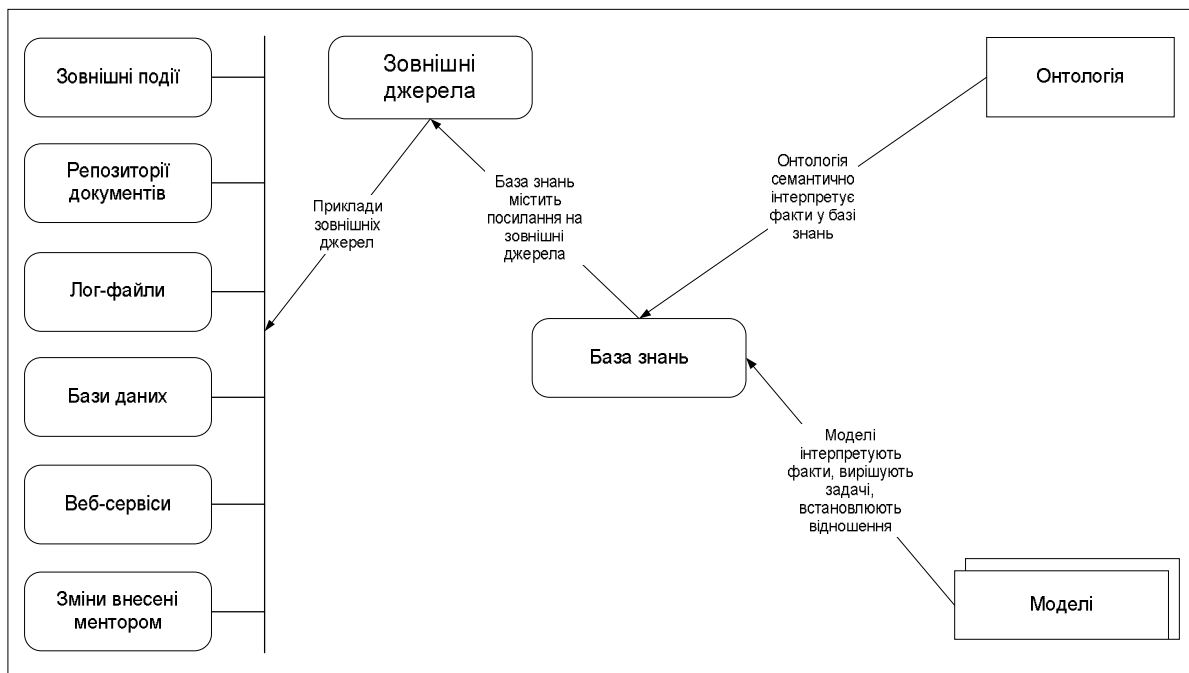


Рис. 1. Структура системи опрацювання знань

Моделі визначають, які задачі розв'язують, у кожен момент у системі і як вони розв'язуються.

Моделі використовують для інтерпретації нових фактів та подій у базі знань. Моделі дають змогу класифікувати об'єкт, семантично інтерпретувати його, встановити відношення з іншими об'єктами.

Посилання на моделі використовують в онтологіях для зберігання складних правил та обмежень, зокрема динамічних обмежень, значення яких залежить від стану бази знань або зовнішнього світу.

Первинним елементом у роботі системи є моделі, зокрема модель координації, яка визначає, які задачі зараз розв'язуються, обирає та активізує моделі для їх розв'язання, координує та планує використання спільних ресурсів, призупиняє виконання низькопріоритетних задач зокрема.

Альтернативою до використання моделі координації є боротьба моделей, що працюють як автономні агенти за ресурси.

Архітектура опрацювання знань  $ArKn$  – це кортеж, що містить базу знань  $BdKn$ , онтологію  $On$ , мережу моделей (схем)  $NMd$ :

$$ArKn = \{BdKn, On, NMd\}$$

База знань  $BdKn$  – це множина фактів про об'єкти та події зовнішнього світу та відношення між ними. Кожен факт та зв'язок є семантично інтерпретованим, тобто його тип визначено в онтології  $On$ :

$$BdKn = \{M(Fc), M(LnFc)\}$$

$$\forall Fc_i : Type(Fc_i) \in On, \forall LnFc_i : Type(LnFc_i) \in On$$

Функція  $Type()$  повертає тип (клас) елемента, визначений в онтології.

Онтологія містить визначення класів  $M(Cl)$ , відношення між ними –  $M(LnCl)$

$$On = \{M(Cl), M(LnCl)\}$$

Кожен клас  $Cl$  визначається набором слотів  $M(SlCl)$ , правил  $M(RuCl)$ , обмежень  $M(CsCl)$ , що визначені на цих слотах, та операцій з класом  $M(OpCl)$ :

$$Cl = \{M(SlCl), M(RuCl), M(CsCl), M(OpCl)\}$$

Кожен слот задано як його множина значень  $M(ValSlCl)$ , правил та обмежень, що діють на рівні слоту –  $M(RuSlCl)$ ,  $M(CsSlCl)$ :

$$SlCl = \{M(ValSlCl), M(RuSlCl), M(CsSlCl)\}$$

Кожен слот має тип, визначений в онтології:

$$\forall SlCl_i : Type(SlCl_i) \in On$$

Для кожного значення слоту  $ValSlCl$  визначено множину допустимих значень  $RgValSlCl$ :

$$\forall ValSlCl_i : ValSlCl_i \in RgValSlCl_i$$

Відношення між класами  $LnCl$  задано на впорядкованій послідовності класів  $Sq(Cl)$ , які воно сполучає:

$$SqLnCl = Sq(Cl)$$

Відношення між класами само є класом у тому розумінні, що визначається набором слотів, правил обмежень та операцій:

$$LnCl = \{SqLnCl, M(SlLnCl), M(RuLnCl), M(CsLnCl), M(OpLnCl)\}$$

Такий підхід дає змогу розглядати відношення як окремі сутності та передбачити можливість формування структур відношень.

Уточнюючи формулу (1), можна сказати, що кожен факт з бази знань відповідає певному класу  $Cl_j$  з онтології і кожне відношення відповідає одному з класів відношень  $LnCl_k$ :

$$\forall Fc_i : Type(Fc_i) = Cl_j \in On \quad \forall LnFc_i : Type(LnFc_i) = LnCl_k \in On$$

Класи онтології  $On$  утворюють ієрархію з використанням відношень. Отже, всю множину відношень можна поділити на відношення успадкування („вертикальні”) та всі інші відношення („горизонтальні”).

$$M(LnCl) = M(VrLnCl) \cup M(HrLnCl)$$

$$M(VrLnCl) \cap M(HrLnCl) = \emptyset$$

Підмножини вертикальних та горизонтальних відношень не перетинаються.

Моделі(схеми) утворюють динамічну мережу що містить множину моделей  $M(Md)$  та множину їх зв'язків  $M(LnMd)$ .

$$NMd = \{M(Md), M(LnMd)\}$$

На відміну від класів онтології, моделі не утворюють чіткої ієрархії і формують динамічну мережу, в якій зв'язки та самі моделі можуть змінюватися, відображаючи процеси навчання, зміни у зовнішньому світі або процес вирішення певної задачі.

Кожна модель може бути в одному з двох станів (активному або пасивному).

$$M(Md) = M(Md_{ac}) \cup M(Md_{ps})$$

$$M(Md_{ac}) \cap M(Md_{ps}) = \emptyset$$

де  $Md_{ac}$ ,  $Md_{ps}$  – активні та пасивні моделі. Активна модель – це модель, ініціалізована інформацією з певного контексту.

Моделі переходять в активний стан на вимогу інших моделей, або при настанні певних подій. Активні моделі використовуються для розв'язання поточних задач системи, інтерпретації знань у системі. Якщо потреба в моделі відпала (отримано результат, досягнуто мету), то модель переходить з активного у пасивний стан.

Зв'язок моделей  $LnMd$  – це відношення активації, яке використовується для прийняття рішення щодо активації моделі (моделей).

Розглянемо процес та структуру взаємодії та активації моделей детальніше. Ініціатором встановлення зв'язку між моделями є модель-активатор. Потреба у встановленні зв'язку виникає не завжди, а тільки тоді, коли для вирішення основної задачі потрібно розв'язати допоміжні задачі, подані в інших моделях. Наприклад, якщо вхідні дані, отримані активатором з контексту, є

неповними і необхідно активізувати інші моделі для довізнання даних. Таке довізнання може полягати в пошуку потрібної інформації в базі даних, Інтернеті, звертанні до консультанта тощо.

Отже, зв'язок  $LnMd$  відповідає певному задач класу  $PrLnMd$ . Такий зв'язок є компонентою багаторазового використання і застосовується для розв'язання подібних задач багатьма різними моделями.  $LnMd$  є моделлю, що розв'язує задачу вибору та активації для визначеного контексту інших моделей з множини альтернатив  $M(MdLnMd)$ .

При цьому послідовно розв'язуються задачі визначення релевантності, оптимального вибору серед релевантних моделей та ініціалізації обраної моделі.

Функція релевантності є відображенням поточного контексту  $Cx$  та множини альтернатив  $M(MdLnMd)$  в множину  $(true, false)$

$$Re LnMd : Cx, M(MdLnMd) \longrightarrow (true, false)$$

Визначення релевантності моделей дає змогу відібрати для процедури вибору тільки релевантні моделі

$$M_{re}(MdLnMd) \subseteq M(MdLnMd),$$

тобто такі моделі,  $Md_{re}$  для яких:

$$Re LnMd(Cx, Md_{re}) = true$$

За відсутності релевантних моделей  $LnMd$  повертає активатору повідомлення про неможливість розв'язання задачі.

Задача оптимального вибору визначає у множині релевантних моделей одну  $Md_{op}$ , застосування якої максимізує функцію вибору  $ChLnMd$  з врахуванням критеріїв вибору  $M(CrChLnMd)$  та контексту  $Cx$ .

$$ChLnMd(Md_{op}, M(CrChLnMd), Cx) \rightarrow \max$$

Операція ініціалізації  $InLnMd$  здійснює відображення поточного контексту у множину значень слотів обраної моделі  $-M(ValSIMd)$ .

$$InLnMd : Cx \rightarrow M(ValSIMd)$$

Підводячи підсумок, визначимо  $LnMd$  як кортеж

$$LnMd = \{Pr LnMd, M(MdLnMd), Re LnMd, ChLnMd, InLnMd\}$$

Процес активації моделі проілюстровано на рис. 2.

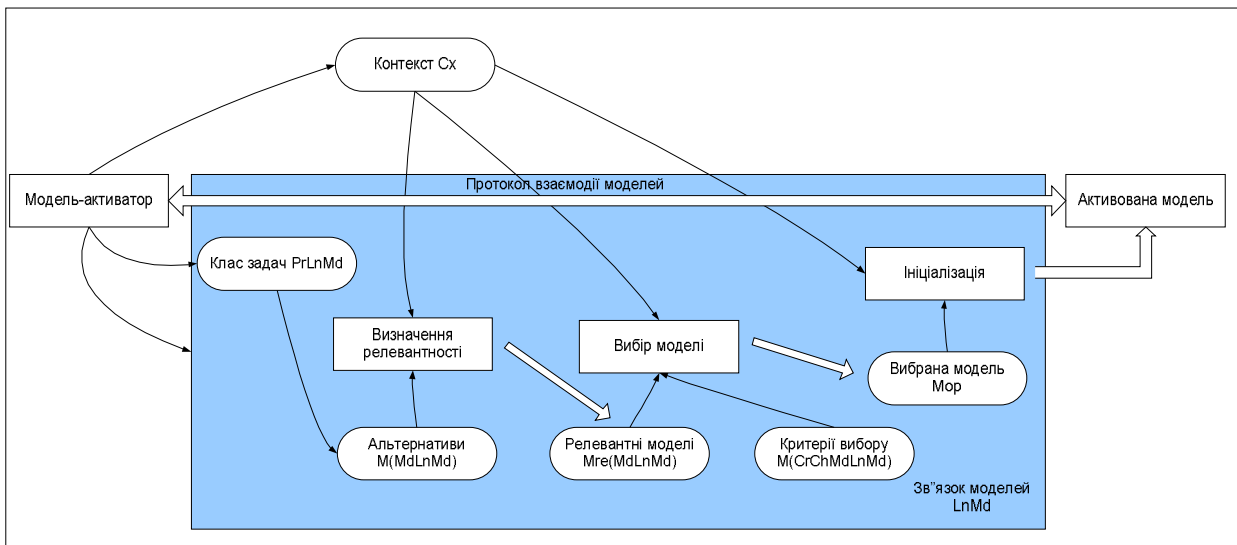


Рис.2. Процес активації моделі



Після активації моделі модель-активатор взаємодіє з активованою моделлю з використанням протоколу взаємодії моделей. Це пов'язано з тим, що активована певним контекстом модель передбачає можливість розв'язання різноманітних задач, і сам процес взаємодії моделей може мати ітераційний характер. Так, наприклад, такими ітераційними задачами можуть бути прогнозування, оцінювання, моделювання, дії. Результат розв'язання однієї задачі є підставою активатору щодо формування наступних запитів до активованої моделі.

Зв'язок моделей  $LnMd$  надає певні послуги та операції до протоколу взаємодії, наприклад, оцінку релевантності, переініціалізацію, оцінку оптимальності застосування моделі. У випадку зміни контексту він проводить оновлення ініціалізації, перевіряє релевантність та оптимальність і так контролює взаємодію активатора та активованої моделі.

Моделю  $Md$  складається зі схеми  $ScMd$  та реалізації  $RIMd$ :

$$Md = \{ScMd, RIMd\}$$

Схема моделі описує її структуру, складові елементи, визначає правила та обмеження на використання моделі, а також перелік можливих операцій та запитів.

Схема є компонентом моделі, який видимий для зовнішнього світу. Вона використовується для взаємодії з цією моделлю.

Схема моделі складається зі слотів  $M(SlScMd)$ , зв'язків між ними  $M(LnSlScMd)$ , правил  $M(RuScMd)$ , обмежень  $M(CsScMd)$ , та операцій  $M(OpScMd)$ :

$$ScMd = \{M(SlScMd), M(LnSlScMd), M(RuScMd), M(CsScMd), M(OpScMd)\}$$

Для кожного слоту визначено множину класів з онтології, об'єктами яких дозволено ініціалізувати слот:

$$RgSlScMd = M_{sl}(Cl) \subseteq M(Cl) \in On$$

Крім того, для слоту визначено його множину значень  $M(ValSlScMd)$ , правил та обмежень, що діють на рівні слоту –  $M(RuSlScMd)$ ,  $M(CsSlScMd)$ , операцій над значеннями слоту  $M(OpSlScMd)$ :

$$SlScMd = \{RgSlScMd, M(ValSlScMd), M(RuSlScMd), M(CsSlScMd), M(OpSlScMd)\}$$

Зв'язок слотів  $LnSlScMd$  специфікується множиною слотів, яку він сполучає  $M(SlLnSlScMd)$ , множиною класів онтології, які використовують для семантичної інтерпретації зв'язку –  $RgLnSlScMd$ , множиною моделей, яку використовують для розуміння та проведення операцій зі зв'язком –  $M(MdLnSlScMd)$ .

$$LnSlScMd = \{M(SlLnScMd), RgLnSlScMd, M(MdLnSlScMd)\}$$

при цьому слоти, що сполучені зв'язком, належать слотам моделі:

$$\forall SlLnScMd \in M(SlLnScMd) : SlLnScMd \in M(SlScMd)$$

Зв'язок моделей відповідає одному з типів горизонтальних зв'язків, визначених в онтології  $On$ :

$$RgLnSlScMd = M_{ln}(LnCl) \subseteq M(HrLnCl) \in On$$

Моделю, що описує зв'язок, є елементом загальної множини моделей:

$$\forall MdLnScMd \in M(MdLnScMd) : MdLnScMd \in M(Md)$$

### Висновок

Запропоновані у роботі підхід та архітектура опрацювання знань у когнітивній інформаційній системі дадуть змогу створювати інтелектуальні системи, які порівняно з існуючими набагато швидше реагують на зміни вимог та бізнес-середовища.

*1. Anne Rozinat. The Need for a Process Mining Evaluation Framework in Research and Practice [Text]/ Anne Rozinat, Ana Karla Alves de Medeiros, Christian W. Gunther, A.J.M.M. Weijters, and Wil M.P. van der Aalst //Business Process Management Workshops. Arthur ter Hofstede Boualem Benatallah Hye-*

Young Paik (Eds.)//Springer-Verlag Berlin Heidelberg, 2008 .-502p.-ISBN 978-3-540-78237-7. 2. Александр Александров. ВІ 2.0: прообраз новой архитектуры бизнес-аналитики [Текст] /Александров А.//Открытые системы.-2007. 3. Буров Є.В. Система моделювання інтелектуальної мережі бізнес-процесів[Текст]/Буров Є.В.// Вісник держуніверситету ' Львівська політехніка' 'Інформаційні системи та мережі'.-2008.- № 610. 4. Cognition. [Electronic resource].-url: <http://www.answers.com/topic/cognition> 5.Peter Voss. Essentials of General Intelligence: The Direct Path to Artificial General Intelligence Engine [Text] / Cassio Pennachin, Ben Goerzel //Artificial General Intelligence. Ben Goertzel, Cassio Pennachin (Eds.).- Springer-Verlag Berlin Heidelberg, 2007. – 517p. 6.Pei Wang. The Logic of Intelligence. /Artificial General Intelligence. Ed. Ben Goerzel, Cassio Pennachin. Springer-Verlag Berlin Heidelberg 2007. 7.Newell, A..Unified Theories of Cognition [Text]/ Newell A.- Harvard University Press.- Reprint edition, ISBN 0-674-92101-1. 8.Cassio Pennachin, Ben Goerzel.Contemporary Approaches to Artificial General Intelligence [Text]./ Cassio Pennachin, Ben Goerzel //Artificial General Intelligence. Eds. Ben Goertzel, Cassio Pennachin.- Springer-Verlag Berlin Heidelberg, 2007. – 517p.9. Cassio Pennachin, Ben Goerzel. The Novamente Artificial Intelligence Engine [Text] / Cassio Pennachin, Ben Goerzel //Artificial General Intelligence. Ben Goertzel, Cassio Pennachin (Eds.).- Springer-Verlag Berlin Heidelberg, 2007. – 517p 10. Hollnagel, Erik, Joint cognitive systems : foundations of cognitive systems engineering [Text] / Erik Hollnagel and David D. Woods.- CRC Press, 2005.- ISBN 0-8493-2821-7.- 241p.11. Anderson J.R. The architecture of cognition. Harward: Harward University Press,- 1984. 12.MIT Encyclopedia of the cognitive sciences. Edited by Robert A. Wilson and Frank C. Keil. A Bradford Book The MIT Press Cambridge, Massachusetts, London, England..- 1999.- ISBN 0-262-73124-X).13. Minsky M. A framework for representing knowledge. /P.Winston ed. The Psychology of computer vision. New York: McGraw-Hill, pp 211-277.14.Bobrow D.G., Winograd T. An overview of KRL-0, a knowledge representation language. Cognitive science. pp 3-46, 1977.15. Gruber T.R. Principles for the design of ontologies used for Knowledge Sharing/ International Journal Human-Computer Studies, 1995.16. Pezzulo G. Schemas and schema-based architectures [Electronic resource].- [http:// www.eucognition.org/wiki/index.php?title=Schemas\\_and\\_Schema-based\\_Architectures](http://www.eucognition.org/wiki/index.php?title=Schemas_and_Schema-based_Architectures). 17. Rumelhart, D.E., & Ortony, A. (1977). The representation of knowledge in memory. In R.C.Anderson, R.J. Spiro, & W.E. Montague (Eds.), Schooling and the acquisition of knowledge (pp. 99-133). Hillsdale, NJ: Lawrence Erlbaum Associates. 18.Keller T. Visualizing Knowledge and Information:an Introduction [Text] /Tanja Keller, Sigmar-Olaf Tergan//Knowledge and Information Visualization. Searching for Synergies.- Lecture notes in computer science.- Springer-Verlag, Berlin Heidelberg, 2005.-378p. 19.Буров Є.В. Автоматизація проектування та моделювання когнітивних бізнес-систем [Текст]: Proceedings of the III International conference on computer science and Information Technologies CSIT 2008/ Буров Є. В.-Lviv: Vezha&Co, 2008